P.1877/86

11-12 1986

informatyka

Systemy wielomikroprocesorowe

Sieć lokalna UMMLAN-2

Komunikacja szeregowa w IBM PC

Systemy ekspertowe

WYDAWNICTWO NOT SIGMA

Cena 120 zł

informatyka

11-12 1986

Nr 11-12

Miesięcznik Rok XXI

Listopad-grudzień 1986

Organ Komitetu Informatyki MNSZWiT oraz Komitetu Naukowo-Technicznego NOT ds. Informatyki

KOLEGIUM REDAKCYJNE:

Dr inż. Wacław ISZKOWSKI, mgr Teresa JABŁOŃSKA (sekretarz redakcji), Władysław KLEPACZ (redaktor naczelny), dr inż. Marek MACHURA, Maria PAWŁAK (sekretarz redakcji), dr inż. Wiktor RZECZKOWSKI, mgr inż. Jan RYZKO, mgr Hanna WŁODARSKA, dr inż. Janusz ZALEWSKI (zastępca redaktora naczelnego)

PRZEWODNICZĄCY RADY PROGRAMOWEJ:

Prof. dr hab. Juliusz Lech KULIKOWSKI

Materiałów nie zamówionych redakcja nie zwraca

Redakcja: 01-507 Warszawa, ul. Mickiewicza 18 m. 17, tel. 39-14-34

Zakł. Graf. „Tamka". Zam. 0779-1300/86. Obj. 4,0 ark. druk. Nakład 8350 egz. P-78.

ISSN 0542-9951, INDEKS 36124

Cena egzemplarza 120 zł
Prenumerata roczna 1440 zł

WYDAWNICTWO NACZELNA ORGANIZACJA TECHNICZNA

SIGMA

CZASOPISM I KSIĄŻEK TECHNICZNYCH

00-950 Warszawa
skrytka pocztowa 1004
ul. Biała 4

W NUMERZE: Strona

Systemy wielomikroprocesorowe (1)
Wojciech Cellary 1

Lokalna sieć komputerowa ETHERNET (2)
Vasilii Zakharov 4

Lokalna sieć komputerowa UMMLAN-2
Krzysztof Zieliński, Jadwiga Indulska, Roman Krasowski 8

Komputery osobiste IBM PC (3)
Waldemar Dworakowski 11

System operacyjny PC-DOS (3)
Roman Grabowicz, Janusz Zalewski 13

Komunikacja szeregowa w mikrokomputerach IBM PC i kompatybilnych
Paweł Maćków 15

Systemy ekspertowe (1)
Jacek Ambroziak 19

Przesłanki i uwarunkowania modelowania konceptualnego
Stanisław Wrycza 22

Z KRAJU 24

MIKROKOMPUTERY '85

ZE ŚWIATA 26

Kto jest kim w IFIP — Robert Piloty (RFN)

TERMINOLOGIA III okł.

Terminologia lokalnych sieci komputerowych (1)

W NAJBLIŻSZYCH NUMERACH:

- Witold Łukaszewicz o przechowywaniu wiedzy i niewiedzy w bazach danych
- Daniel Tabak o mikroprocesorze INTEL 80386 i nowych mikroprocesorach 32-bitowych
- Wojciech Mokrzycki o systemie programowania grafiki komputerów PSG 2.2
- Marek Pawłowski o programatorze pamięci UVEPROM serii 27XXX
- Jan Bielecki o kompilatorze języka C

WOJCIECH CELLARY
Instytut Automatyki
Politechnika Poznańska

POLSKIE TOWARZYSTWO INFORMATYCZNE

P. 1877/86

Pierwsza wersja artykułu została przedstawiona podczas Szkoły Jesiennej PTI „Współczesne kierunki rozwoju informatyki" — Rydzyna, 9—13 października 1984 r.

Systemy wielomikroprocesorowe (I)

W rozwoju systemów komputerowych szczególnie wyraźnie ujawnia się naturalne dążenie do konstrukcji systemów o coraz większej efektywności działania. Zwiększenie tej efektywności uzyskuje się z jednej strony przez doskonalenie technologii układów składających się na system komputerowy, w szczególności na jego procesor (warto tu uzmysłowić sobie odbytą w ciągu czterdziestu lat drogę od lamp elektronowych do układów VLSI), a z drugiej — przez doskonalenie architektury systemów komputerowych, polegające na zwiększeniu stopnia równoległości operacji realizowanych przez system. Można mówic o kilku poziomach tak rozumianego doskonalenia architektury systemu. Pierwszym z nich jest poziom architektury wewnętrznej procesora. Przykładem doskonalenia architektury systemu na tym poziomie jest wprowadzenie zakładkowania, pozwalającego na zrównoleglenie fazy wykonywania jednej mikroinstrukcji z fazą pobierania następnej, a także wprowadzenie arytmometru z przewidywaniem przeniesienia, umożliwiającego równoległe wykonywanie operacji arytmetycznych na wszystkich bitach słowa.

Drugim poziomem doskonalenia architektury systemu jest poziom klasycznego komputera. Przykładem może tu być wprowadzenie autonomicznych kanałów wejścia-wyjścia, umożliwiających równoległe działanie procesora centralnego i wykonywanie operacji wejścia-wyjścia. Szczególną rolę w doskonaleniu architektury systemu na tym poziomie odegrało wprowadzenie przerwań jako środka informowania procesora centralnego o zajściu zdarzeń asynchronicznych w stosunku do jego działania, w szczególności — o zakończeniu wykonywania równoległych operacji wejścia-wyjścia.

Trzecim, najwyższym poziomem jest poziom systemu. Doskonalenie architektury na tym poziomie odbywa się przez wprowadzenie równolegle działających i komunikujących się między sobą procesorów centralnych, które nie muszą być identyczne.

Architektura wieloprocesorowa systemów komputerowych ma, oprócz wspomnianej możliwości zwiększania efektywności systemu, jeszcze dwie inne potencjalne zalety. Pierwszą z nich jest możliwość zwiększenia niezawodności systemu, na przykład wskutek równoległej realizacji tych samych funkcji przez kilka procesorów z okresowym porównywaniem wyników, a także dzięki testowaniu i diagnostyce jednych procesorów (ściślej — całych modułów) przez drugie, dzięki przejmowaniu funkcji uszkodzonych procesorów przez inne sprawne itp. Drugą potencjalną zaletą jest możliwość modularyzacji systemu zgodnie z wymaganiem danego zastosowania i uzyskanie dzięki temu dużej elastyczności, co ma szczególne znaczenie w wypadku jego modyfikacji i rozwoju.

Zalety wieloprocesorowej architektury systemów komputerowych dostrzeżono już na dość wczesnym etapie rozwoju tych systemów. Pierwszymi eksperymentalnymi systemami o takiej architekturze były systemy C.mmp (computer. multi-mini-procesor), którego budowę rozpoczęto w 1971 roku, oraz Pluribus, którego budowę rozpoczęto w 1972 roku. W systemach tych wykorzystywano technologię, a nawet gotowe moduły minikomputerowe.

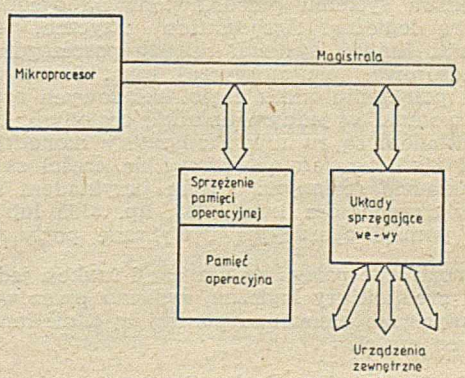
Jednakże, dopiero rozwój szesnastobitowych mikroprocesorów trzeciej generacji stworzył techniczne i ekonomiczne warunki do powszechnego stosowania systemów o architekturze wieloprocesorowej, w tym wypadku — systemów wielomikroprocesorowych. Przez systemy wielomikroprocesorowe rozumie się w tym artykule systemy wielomikroprocesorowe silnie powiązane (ang. tightly (closely) coupled microcomputer systems), tj. takie, w których komunikacja między mikroprocesorami odbywa się przez wspólną pamięć operacyjną. W przeciwieństwie do systemów silnie powiązanych, istnieją jeszcze systemy luźno powiązane (ang. loosely coupled microprocessor systems), w których komunikacja między mikroprocesorami odbywa się przez porty wejścia-wyjścia.

Celem artykułu jest wprowadzenie do problematyki systemów wielomikroprocesorowych, ukazanej na tle omówionych pokrótce systemów jednomikroprocesorowych.

ARCHITEKTURA SYSTEMÓW JEDNOMIKROPROCESOROWYCH

Schemat architektury systemów jednomikroprocesorowych przedstawiono na rysunku 1. W systemie takim można wyróżnić trzy główne części składowe: mikroprocesor, pamięć operacyjną oraz układy sprzęgające we-wy. Części te komunikują się między sobą przez magistralę, będącą zbiorem odpowiednich linii sygnałowych. Z funkcjonalnego punktu widzenia magistrala systemu mikrokomputerowego składa się z magistrali sterującej, magistrali adresowej

Doc. dr hab. inż. WOJCIECH CELLARY ukończył w 1974 r. studia na Wydziale Elektronicznym Politechniki Poznańskiej. W 1977 r. obronił pracę doktorską, a w 1981 r. uzyskał habilitację w dziedzinie informatyki. Jest pracownikiem Zakładu Badań Operacyjnych i Systemów Komputerowych Instytutu Automatyki Politechniki Poznańskiej. Jego zainteresowania zawodowe obejmują modelowanie i ocenę działania systemów komputerowych, systemy rozproszonych baz danych oraz systemy mikrokomputerowe. Obecnie pełni funkcję wiceprezesa Zarządu Głównego Polskiego Towarzystwa Informatycznego.



Rys. 1. Architektura systemów jednomikroprocesorowych

i magistrali danych. Zasadą działania systemu wielomikroprocesorowego jest synchroniczna praca na magistrali, co oznacza, że sygnały na magistrali mogą pojawić się wyłącznie w ściśle określonych momentach wyznaczonych przez tzw. cykl zegarowy. Przykładowo, pobranie zawartości komórki pamięci operacyjnej przez mikroprocesor przebiega następująco: mikroprocesor wysyła adres żądanej komórki na magistralę adresową, a po odczekaniu odpowiedniej liczby cykli zegarowych odczytuje z magistrali dane. Zadaniem pamięci operacyjnej jest wysłanie — po zdekodowaniu adresu — zawartości zaadresowanej komórki na magistralę danych w odpowiednim momencie i na odpowiednio długi czas. Jeśli pamięć operacyjna jest zbyt wolna w stosunku do szybkości pracy mikroprocesora i nie może wysłać danej w odpowiednio krótkim czasie, to informuje o tym fakcie mikroprocesor, wysyłając odpowiedni sygnał (WAIT) przez magistralę sterującą. Mikroprocesor wydłuża wówczas okres oczekiwania na daną o dodatkowy cykl zegarowy i w tak określonym momencie odczytuje ją z magistrali danych. Jak widać, wymiana informacji między częściami składowymi systemu mikrokomputerowego odbywa się w ściśle określonych momentach i bez wymiany potwierdzeń.

Opisany cykl pracy mikroprocesora nazywa się cyklem maszynowym. Jest on związany ze zmianą adresu przesyłanego przez magistralę adresową. Cykle maszynowe nie są jednakowej długości, choć zawsze są wielokrotnością cyklu zegarowego.

Oczywiście, cykl maszynowy zapisu do pamięci operacyjnej jest analogiczny do opisanego cyklu maszynowego odczytu z pamięci. Mikroprocesor wysyła adres komórki na magistralę adresową oraz następnie daną na magistralę danych. Po upływie określonego czasu kończy cykl maszynowy usuwając sygnały, a więc zakładając, że dane zostały przyjęte przez pamięć operacyjną. Podobnie jak poprzednio, pamięć może żądać wydłużenia cyklu maszynowego. Jedyna różnica między cyklem maszynowym odczytu i zapisu polega na wysłaniu na magistralę sterującą odpowiedniego sygnału kierunku transmisji.

Komunikacja między mikroprocesorem a układami sprzęgającymi we-wy odbywa się w postaci analogicznych cykli maszynowych. Jeśli adres takiego układu jest zawarty w przestrzeni adresowej pamięci, to przebieg cykli jest identyczny, gdyż układ jest traktowany tak jak komórki pamięci operacyjnej. Jeśli natomiast adres układu sprzęgającego jest zawarty w przestrzeni adresowej we-wy, to różnica polega na użyciu odpowiedniego sygnału na magistrali sterującej, informującego o adresowaniu tej przestrzeni.

Komunikacja między układami sprzęgającymi we-wy a pamięcią operacyjną także przebiega w postaci podobnych cykli maszynowych. Komunikacja taka zachodzi w wypadku transmisji przez kanał DMA — bezpośredniego dostępu do pamięci operacyjnej. Na czas transmisji kanał DMA przejmuje sterowanie magistralą, wymuszając przejście mikroprocesora w stan dużej impedancji. Przejście to odbywa się po zakończeniu aktualnego cyklu maszynowego mikroprocesora.

Na zakończenie omawiania cykli pracy magistrali systemu mikrokomputerowego warto wspomnieć o tzw. cyklu rozkazowym, obejmującym wykonanie pełnego rozkazu mikroprocesora. Cykl rozkazowy składa się z pewnej liczby cykli maszynowych, zależnej od formatu rozkazu.

Architektura systemu przedstawiona na rysunku 1 jest architekturą logiczną. Architektura fizyczna, rozumiana jako podział funkcji systemu między poszczególne układy mikroprocesorowe, jest zazwyczaj inna. Jedna możliwość polega na połączeniu kilku części składowych architektury logicznej systemu w jednym układzie scalonym. Ekstremalnym przykładem są tu mikrokomputery monolityczne, np. serii Intel 8048, zawierające wszystkie części składowe w jednym układzie scalonym. Innym przykładem są układy rodziny mikroprocesora Intel 8085, np. 8155 lub 8755, zawierające pamięć operacyjną i wybrane porty we-wy.

Druga możliwość polega na realizacji funkcji jednej części składowej architektury logicznej systemu przez wiele układów scalonych, co omówiono poniżej w odniesieniu do poszczególnych części.

Mikroprocesor rzadko występuje w systemie mikrokomputerowym bez pewnych układów pomocniczych. Takimi układami są dla mikroprocesora Intel 8080 — układ zegarowy 8224 i sterownik magistrali sterującej 8228, dla mikroprocesora Intel 8085 — rejestr zatrzaskowy np. 8212, dla mikroprocesorów Intel 8088 i Intel 8086 — układ zegarowy 8284 oraz sterowniki magistrali adresowej (rejestry zatrzaskowe) 8282 lub 8283, magistrali danych 8286 lub 8287 i magistrali sterującej 8288. Istnienie tych układów wynika z faktu, że z technologicznego punktu widzenia jednym z najbardziej ograniczonych, a przez to najcenniejszych, zasobów mikroprocesora jako układu scalonego są jego końcówki. Ponieważ sygnałów jest więcej niż końcówek układu scalonego, jednej końcówce należy przypisać kilka sygnałów. Istnieje kilka rozwiązań tego problemu. Pierwsze polega na przetwarzaniu kilku sygnałów z magistrali sterującej i wypracowywaniu przez układ pomocniczy jednego sygnału wejściowego dla mikroprocesora. Funkcję taką spełniają układy zegarowe. Drugie rozwiązanie polega na wysyłaniu niektórych sygnałów sterujących w postaci zakodowanej i rozkodowywaniu ich w układzie pomocniczym. Funkcję taką spełniają sterowniki magistrali sterującej. Trzecie rozwiązanie polega na multipleksowaniu sygnałów, czyli wykorzystywaniu jednej końcówki mikroprocesora do przesyłania różnych sygnałów rozdzielonych w czasie. Najczęściej stosuje się multipleksowanie adresów i danych. Rozwiązanie to wymaga stosowania rejestrów zatrzaskowych do przechowania i udostępniania na magistrali — na odpowiednio długi czas — sygnału pojawiającego się w pierwszej kolejności. Czwarte rozwiązanie polega na sprzętowym programowaniu (przez podanie stałego sygnału na określoną końcówkę mikroprocesora) zestawu sygnałów mikroprocesora. Rozwiązanie takie jest stosowane — na przykład — w mikroprocesorach Intel 8088 i Intel 8086 do określania trybu pracy mikroprocesora.

Pamięć operacyjna systemów mikroprocesorowych dzieli się na pamięć wyłącznie odczytywalną EPROM oraz pamięć zapisywalną RAM. Istnienie pamięci EPROM w systemie jest niezbędne ze względu na to, że pamięć RAM wykonana w obecnej technologii traci swą zawartość po wyłączeniu zasilania. Pamięć EPROM jest więc konieczna do poprawnego wznowienia działania systemu.

Sprzężenie pamięci operacyjnej stanowi dekoder adresów oraz rejestry buforowe. W niektórych wypadkach stosuje się również rejestry segmentowe umożliwiające rozszerzenie przestrzeni adresowej mikroprocesora. W wypadku stosowania pamięci dynamicznej RAM, układami pomocniczymi tej pamięci są układy odświeżania zawartości.

Do układów sprzęgających we-wy zalicza się wszystkie te układy mikroprocesorowe, które pośredniczą w komunikacji mikroprocesora z urządzeniami zewnętrznymi. Należą do nich: port we-wy szeregowego 8251, port we-wy równoległego 8255, układy programowalnych liczników — zegarów 8253 i 8254, kanał DMA 8257, sterownik przerwań 8259, sterowniki pamięci na dyskach elastycznych 8271 i 8272, sterownik monitora ekranowego 8275 i inne.

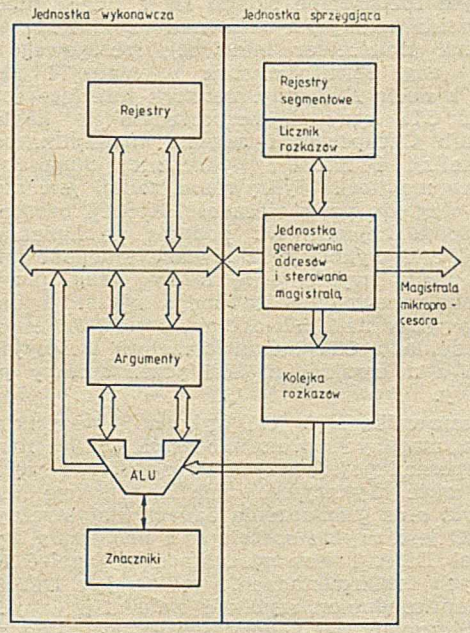
ARCHITEKTURA WEWNĘTRZNA MIKROPROCESORA INTEL 8086

Problemy architektury systemów wielomikroprocesorowych muszą być rozważane na dwóch poziomach: na poziomie architektury wewnętrznej mikroprocesorów oraz na poziomie architektury systemu. W tej części artykułu zostanie przedstawiona architektura wewnętrzna mikroprocesorów zaprojektowanych z myślą o pracy w systemach wielomikroprocesorowych, na przykładzie mikroprocesora szesnastobitowego Intel 8086. Wybór tego mikroprocesora nie jest przypadkowy, gdyż ze wszystkich mikroprocesorów szesnastobitowych ma on największe szanse rozpowszechnienia w Polsce w najbliższej przyszłości.

W celu przedstawienia problemów architektury wewnętrznej takich mikroprocesorów, warto powrócić do przypadku systemów jednomikroprocesorowych z mikroprocesorem drugiej generacji (np. Intel 8080), wyposażonych w kanał DMA. Teoretycznie system taki może być uznany za ograniczony system wielomikroprocesorowy. Kanał DMA można bowiem uznać za specjalizowany mikroprocesor, z tego względu, że może on samodzielnie ubiegać się o dostęp do magistrali. Uaktywnienie kanału DMA powoduje zawieszenie działania mikroprocesora; praca jednego układu wyklucza więc pracę drugiego — równoległość ich działania jest zerowa. Wniosek taki jest sprzeczny z podstawowym celem budowy systemów wielomikroprocesorowych, jakim jest zwiększenie efektywności systemu przez równo-

ległą realizację jego funkcji. Może powstać w tym miejscu pytanie o celowość stosowania kanałów DMA w systemach jednomikroprocesorowych? Aby nie pozostawiać tego pytania bez odpowiedzi, należy wyjaśnić, że transmisja do (z) pamięci operacyjnej za pomocą kanału DMA jest wielokrotnie szybsza niż analogiczna transmisja za pomocą mikroprocesora. Wynika to z faktu, że w każdym cyklu maszynowym kanału DMA przesyłane są do (z) pamięci operacyjnej tylko dane. Natomiast w wypadku transmisji za pomocą mikroprocesora, na jeden cykl maszynowy przeznaczony na przesłanie transmitowanych danych przypada wiele cykli związanych z wykonywaniem programu. Ponadto, jeśli transmisja przez kanał DMA nie jest na tyle szybka, aby wykorzystywać wszystkie kolejne cykle maszynowe magistrali, to kanał DMA zwalnia magistralę, umożliwiając wznowienie pracy mikroprocesorowi, na przykład, na jeden cykl. W ciągu całej transmisji możliwa jest zatem pewna równoległość pracy mikroprocesora i kanału DMA.

Powyższe wyjaśnienia nie zmieniają faktu, że gdyby na podobnej zasadzie połączyć w systemie dwa mikroprocesory drugiej generacji, to równoległość ich pracy byłaby minimalna. Z tego względu, dla mikroprocesora trzeciej generacji, jakim jest Intel 8086, opracowano inną architekturę wewnętrzną, przedstawioną na rysunku 2.



Rys. 2. Architektura wewnętrzna mikroprocesora Intel 8086

Mikroprocesor Intel 8086 jest zbudowany z dwóch jednostek, pracujących asynchronicznie względem siebie: jednostki wykonawczej (ang. Execution Unit) oraz jednostki sprzęgającej z magistralą (ang. Bus Interface Unit). Jednostka sprzęgająca realizuje całą współpracę mikroprocesora magistrali. Jednostka wykonawcza realizuje rozkazy mikroprocesora, a w razie potrzeby dostępu do magistrali (w celu pobrania rozkazu lub argumentu, albo zapisania wyniku) komunikuje się z jednostką sprzęgającą, sygnalizując jej swoje żądanie.

Jednostka sprzęgająca jest wyposażona w wewnętrzną kolejkę, do której pobiera rozkazy z pamięci operacyjnej, wyprzedzająco w stosunku do żądań jednostki wykonawczej. Dzięki temu, w sytuacji, w której mikroprocesor nie ma dostępu do magistrali ze względu na jej przydział innemu mikroprocesorowi w systemie wielomikroprocesorowym, praca jednostki wykonawczej nie jest zatrzymywana, gdyż rozkazy są pobierane przez nią z wewnętrznej kolejki jednostki sprzęgającej. Oczywiście nie wyklucza się sytuacji, w których jednostka wykonawcza żądając pobrania rozkazu przez jednostkę sprzęgającą natrafia na pustą kolejkę wewnętrzną i niedostępną magistralę, i musi czekać. Sytuacja taka może pojawić się — na przykład — po wykonaniu rozkazu skoku, gdy kolejka wewnętrzna jednostki sprzęgającej musi być wyzerowana.

dokończenie na str. 7

Międzynarodowa
Wystawa Komputerowa
INFOSYSTEM '87

Wrocław, 7—11 kwietnia 1987

Oprogramowanie i sprzęt dla:

- **systemów uniwersalnych ogólnego przeznaczenia**
- **systemów problemowo-zorientowanych na banki, transport, handel, budownictwo, służbę zdrowia, rolnictwo, turystykę**
- **systemów wspomagania prac B-R oraz prac konstrukcyjnych i projektowych**
- **systemów wspomagania wytwarzania i zarządzania**
- **systemów teleprzetwarzania i terminali ogólnego zastosowania oraz terminali specjalizowanych**
- **komputerów personalnych**
- **komputerów edukacyjnych i domowych**
- **systemów automatyzacji prac biurowych**

oraz:

- **kasy i wagi elektroniczne**
- **kalkulatory elektroniczne**
- **akcesoria i materiały eksploatacyjne dla techniki komputerowej**
- **książki, czasopisma i podręczniki z dziedziny informatyki**

Międzynarodowe Targi Poznańskie

ul. Głogowska 14,

60-734 Poznań

tel. 69-93-41,

telex: 0413251 targ pl

EO/21/K/87

VASILII ZAKHAROV
Uniwersytet Genewski
Genewa
Szwajcaria

Lokalna sieć komputerowa ETHERNET (2)

W drugiej części artykułu omówiono sprzęganie urządzeń i komputerów obliczeniowych z siecią Ethernet oraz — układy LSI przeznaczone do tego celu. Przedstawiono także wzajemne sprzęganie sieci przez pomosty i bramy oraz opisano przykładową realizację sieci Ethernet na Uniwersytecie Genewskim

Sprzęganie urządzeń

Problem sprzęgania urządzeń z siecią jest dość złożony ze względu na różnorodność komputerów macierzystych, z którymi sprzęgane są urządzenia. Jednakże, samo zagadnienie sprzęgania komputerów z siecią zostanie omówione w następnym punkcie.

Terminale, np. terminale ASCII lub ich emulatory, i drukarki, łączy się z siecią Ethernet za pomocą specjalnych jednostek sprzęgających, tzw. NIU (ang. network interface units), lub stacji obsługi. Jednostki sprzęgające oparte na mikroprocesorach, wytwarzane już przez wielu producentów, od strony sieci łączą się przez kabel połączeniowy z nadajnikiem-odbiornikiem, a z drugiej strony mają po 32 porty dostępne dla urządzeń. Typowe porty do łączenia urządzeń są zgodne z normą RS-232C (maksymalna szybkość — 38,4 kb/s) lub RS-422. Oprogramowanie dla jednostek sprzęgających obejmuje przede wszystkim protokoły Bisync i SDLC, ale także — TCP/IP i protokoły ISO oraz XNS. Dla środowisk wymagających rozbudowy istnieją także jednostki sprzęgające do magistrali Multibus i IEEE-488.

Stacje obsługi terminali są oczywiście o wiele bardziej złożone niż zwykłe jednostki sprzęgające. Zapewniają one pewien rodzaj zarządzania siecią i obsługi nazw, umożliwiając definiowanie nazw logicznych, gromadzenie danych statystycznych, sterowanie rozgłaszaniem komunikatów i dostępem do nich. Pozwalają także na programowe ustawianie różnych parametrów, np. szybkości transmisji, kontroli echa itp.

Urządzenia inteligentne mogą być także sprzęgane z siecią Ethernet przez stacje obsługi terminali, lecz coraz częściej komputery osobiste lub indywidualne stanowiska robocze łączone są z siecią bezpośrednio za pomocą płyt sprzęgających. Dotyczy to w szczególności komputerów pracujących pod nadzorem systemu operacyjnego Unix BSD4.2, którego integralną częścią jest protokół TCP/IP. Przykładowo, dla komputerów IBM PC dostępnych jest już wiele płyt sprzęgających je z siecią Ethernet opartą na cienkim kablu RG58. Ostatnio pojawiły się w sprzedaży płyty zawierające inteligencję, umożliwiające prawie całkowite odciążenie komputera od obsługi sieci (do warstwy transportowej włącznie).

W celu sprzęgania z siecią Ethernet wszelkich innych urządzeń, nie mających charakteru obliczeniowego, zaleca się stosowanie jednostek inteligentnych lub stacji obsługi, pośredniczących w połączeniu z siecią. Obecnie istnieją już stacje obsługi drukarek, umożliwiające użycie bardziej złożonych drukarek laserowych, stacje obsługi plików, pozwalające na współdzielenie pamięci dyskowej, itp. Innym przykładem tego podejścia jest automatyzacja laboratorium, gdzie standardową magistralę, jak np. VME, CAMAC lub Multibus, sprzęga się z siecią za pomocą komputera osobistego lub stanowiska roboczego. Odpowiednie jednostki sprzęgające, wraz z oprogramowaniem wysokiego poziomu, istnieją już niemal dla wszystkich kombinacji połączeń bardziej znanych magistrali i komputerów osobistych lub stanowisk roboczych.

Mikroukłady sprzęgające

Najważniejszym aspektem rozwoju sieci Ethernet jest handlowa dostępność mikroukładów sprzęgających. W przeciwnym razie zrealizowanie sprzężenia z siecią byłoby nie tylko bardziej kosztowne, lecz także znacznie trudniejsze. Dzięki dostępności mikroukładów sprzęgających dla sieci zgodnej z normą IEEE 802.3, wyprzedza ona znacznie pod względem rozwoju inne lokalne sieci komputerowe.

Tradycyjne rozwiązanie składa się z trzech mikromodułów, sprzężenia nadajnika-odbiornika z kablem, tzw. CTI (ang. coaxial transceiver interface), szeregowego sprzęgu sieci, SNI (ang. serial network interface) i sterownika sprzężenia z siecią NIC (ang. network interface controller). Mikroukład CTI łączy się bezpośrednio z kablem sieci Ethernet lub Cheapernet i zawiera: nadajnik, odbiornik, detektor kolizji i monitor nadmiernej długości pakietu (ang. jabber monitor). Wykrywanie kolizji jest szczególnie trudne, ze względu na konieczność detekcji bardzo małych zmian poziomu napięcia DC podczas wystąpienia kolizji. Mikroukład SNI wykonuje kodowanie i dekodowanie bifazowe i zawiera zwykle pętlę fazową (ang. phase-locked loop) w celu zapewnienia stabilnego sygnału synchronizującego. Klasyczny nadajnik-odbiornik sieci Ethernet składa się z mikroukładu CTI i transformatora izolacyjnego. Łączy się on z mikroukładem SNI przez opisany już kabel połączeniowy.

Celem sterownika NIC jest wykonanie wszystkich funkcji łącza wymaganych w normie IEEE 802.3 i rozpoznanie adresów (operacji pojedynczych i rozgłaszania częściowego). Może on także wykazywać niektóre funkcje testowania i diagnostyki oraz gromadzenia danych statystycznych. Sterownik NIC jest zwykle sprzężony z magistralą systemową przez kanał DMA i wewnątrzukładową pamięć FIFO. W nowoczesnych mikroukładach, takich jak DP8390 firmy National Semiconductor, szybkość transmisji danych osiąga 10 Mb/s. Sterownik NIC łączy się bezpośrednio z mikroukładem SNI, lecz do tego celu wymaga zwykle dodania kilku innych układów.

Istnieje kilku producentów półprzewodników, którzy wytwarzają mikroukłady kodowania bifazowego i sterowniki łącza dla sieci Ethernet (zwane ogólnie mikroukładami LANCE, ang. LAN Controller for Ethernet). Jednakże ostatnio, np. firmie Seeq, udało się zredukować liczbę niezbędnych mikroukładów do dwóch (por. tabela). W mikroukładzie 8004 scalono funkcje sterowania łączem, kodowania bifazowego i wielokrotnego buforowania przy odbiorze.

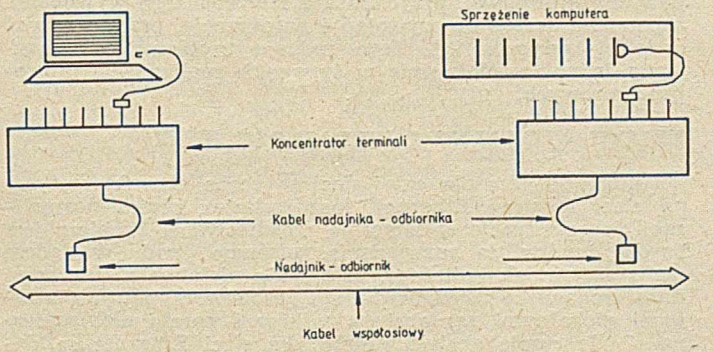
Wybrane mikroukłady LSI dla sieci Ethernet (J.Z.)

Producent	Opis mikroukładów
Intel	Sterownik łącza 82586, koder-dekoder bifazowy 82501
AMD	Sterownik łącza Am 7990, koder-dekoder bifazowy Am 7991, nadajnik-odbiornik Am 7995
Rockwell	Sterownik łącza R68802
Fujitsu	Sterownik łącza MB8795, koder-dekoder bifazowy MB502A
National Semiconductor	Sterownik łącza DP8390, koder-dekoder bifazowy DP8391, nadajnik-odbiornik DP8392
Mostek	Sterownik łącza MK68590, koder-dekoder bifazowy MK3891
Seeq Technology	Sterownik łącza 8003, koder-dekoder bifazowy 8002

Sprzęganie komputerów macierzystych

Sprzęganie komputerów macierzystych z siecią Ethernet jest bardzo złożone, ponieważ obejmuje nie tylko sprzęganie sprzętu, ale także dostosowanie protokołów, wybór zasad ruchu i zintegrowanie ich z systemem operacyjnym komputera. Innym istotnym elementem, często nie wymienianym, jest obciążenie obliczeniowe komputera macierzystego wynikające z obsługi ruchu w sieci.

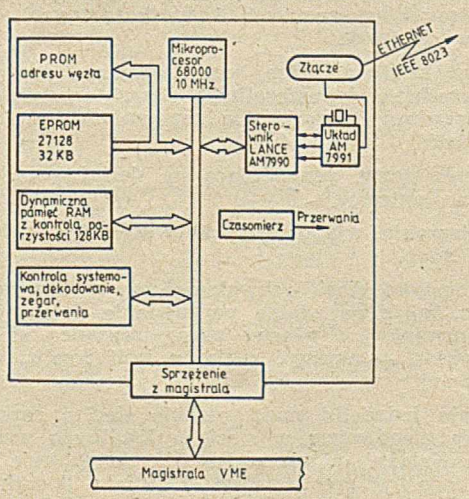
Ze względu na brak miejsca, w niniejszym artykule omówiono tylko trzy główne metody sprzęgania komputerów macierzystych. Pierwsza i najprostsza metoda umożliwia dostęp terminalowy, do którego używa się bez zmiany standardowych metod sprzęgania terminali i standardowych systemów operacyjnych. W tym wypadku po stronie komputera macierzystego dodaje się jedynie jednostkę sprzężenia sieciowego, NIU, tę samą, która służy do sprzęgania z siecią samych terminali. Przykładem takiej jednostki jest CS/1 firmy Bridge Communications, przez którą można dołączyć 32 terminale (rys. 1).



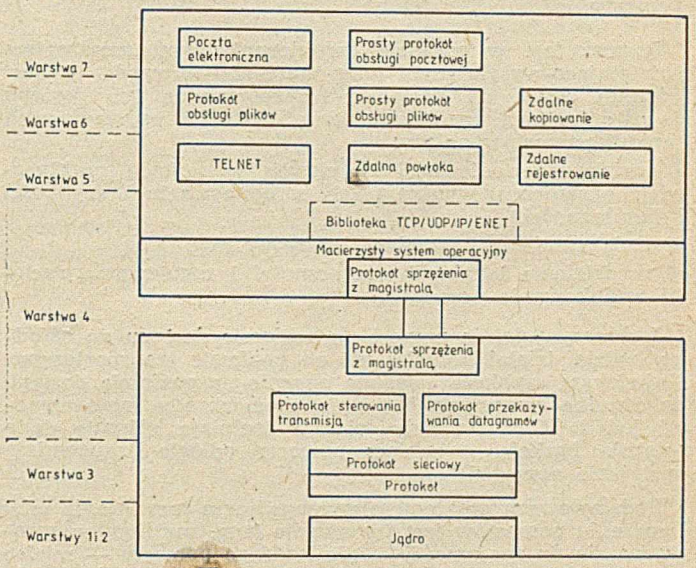
Rys. 1. Sprzężenie terminala i komputera macierzystego

Wadą tej metody sprzęgania jest jej duży koszt i możliwość dołączenia przez każdą jednostkę NIU ograniczonej liczby komputerów. Metoda ta nie jest też zbyt dogodna do przesyłania plików. Bardziej skuteczna metoda polega na sprzęganiu standardowych portów sieciowych komputera za pomocą specjalnego sprzętu dostosowanego do protokołu SNA, DECNET lub X.25. Ta technika jest znacznie bardziej użyteczna i dostosowana do przesyłania plików, lecz ma także ograniczenia, wynikające z niezbyt dobrego dopasowania sprzętu i oprogramowania do ruchu w sieciach lokalnych.

Nawiązując do wymienionego urządzenia sprzęgającego NIU, warto zauważyć, że obecnie istnieją już sprzężenia prawie do wszystkich bardziej znanych magistrali standardowych. Przykładowo, amerykańska firma CMC (Communications Machinery Corp., Santa Barbara, stan Kalifornia) wytwarza zestaw płyt sprzęgających z siecią Ethernet, które umożliwiają dołączenie magistrali Versabus, VME, Multibus, Unibus, Q-bus, a także magistrali IBM PC/AT. Ponadto, dla niektórych komputerów macierzystych dostępne są moduły programowe, zapewniające współpracę przez protokoły TCP/IP lub XNS. Sprzężenie sprzętowe od strony komputera macierzystego jest widoczne jako pamięć na magistrali, a programowe — jako program obsługi we-wy (ang. I/O driver). Przykładowe rozwiązania przedstawiono na rys. 2 i 3.



Rys. 2. Sprzężenie magistrali VME



Rys. 3. Struktura oprogramowania dla sprzężenia magistrali VME

Trzecia metoda sprzęgania komputerów macierzystych z siecią Ethernet polega na użyciu specjalnie zaprojektowanych sprzężeń sprzętowych i programowych. Można ją zilustrowac dwoma przykładami. W pierwszym rozwiązaniu, dla komputera VAX z systemem operacyjnym VMS, stosuje się płytę sprzęgającą firmy Interlan i oprogramowanie według protokołu TCP/IP firmy Wollongong. Drugi przykład dotyczy sprzęgania komputerów IBM. W rozwiązaniu firmy Fibronics sprzężenie sprzętowe (sterownik sieci Ethernet K200) łączy się z kanałem multipleksera i przez nadajnik-odbiornik z siecią Ethernet, zapewniając maksymalną przepustowość powyżej 2,5 Mb/s. Oprogramowanie związane ze sterownikiem K200, tzw. KNET, zapewnia współpracę według protokołu TCP/IP, zdalne rejestrowanie się (ang. remote logon) według protokołu Telnet i przesyłanie plików według TFTP. Oprogramowanie KNET zrealizowano jako maszynę wirtualną w systemie operacyjnym VM/SP, a więc bez żadnych modyfikacji tego systemu.

STRUKTURA SIECI ETHERNET

W pierwszej części artykułu stwierdzono, że maksymalną długość jednego segmentu kabla, 500 m, można zwiększyć do ok. 2800 m, sprzęgając sąsiednie segmenty za pomocą repeterów. Nie nakłada się żadnych ograniczeń na strukturę sieci, za wyjątkiem ograniczenia liczby kaskadowo połączonych repeterów i nie dopuszcza się do tworzenia zamkniętych pętli. Odległość między dwoma sąsiednimi segmentami kabla jest więc równa długości dwóch kabli połączeniowych, tzn. 100 m, lecz można ją zwiększyć.

W tym celu stosuje się tzw. odległe repetery, których funkcja pozostaje nie zmieniona, lecz rozkłada się na dwie jednostki. Przez dodanie odpowiednich mikromodułów sterowania linią (ang. medium drivers) obie jednostki można rozdzielić na dowolną odległość nie przekraczającą ograniczenia odpowiadającego minimalnemu czasowi przesyłania pakietu, tj. 51,2 μs. Rozdzielone jednostki mogą być połączone dowolnym ośrodkiem dopuszczającym szybkość transmisji 10 Mb/s, lecz zazwyczaj stosuje się włókna światłowodowe o typowej długości 1 km. Takie odległe repetery zwiększają znacznie elastyczność konfiguracji sieci lokalnej.

Powstaje jednak pytanie, co robić, jeśli sieć powinna obejmować większy obszar geograficzny lub przewidywane natężenie ruchu przekracza możliwości jednej sieci Ethernet (lub jej pięciu segmentów). W takim wypadku istnieją dwa rozwiązania. Jedno z nich polega na wzajemnym połączeniu kilku sieci Ethernet w jednorodną strukturę, a drugie — na sprzężeniu jednej lub kilku sieci Ethernet z inną siecią zbudowaną według odmiennych zasad. W wypadku sieci jednorodnych, jednostkę sprzęgającą dwie sieci

Ethernet nazywa się pomostem (ang. bridge, internet router), natomiast w drugim wypadku — bramą (ang. gateway). Podobnie jak repetery, wyróżnia się pomosty lokalne i odległe, a jedyna różnica polega na braku ograniczenia odległości między rozdzielonymi pomostami.

Pomosty

Pomost jest w zasadzie dwukierunkowym urządzeniem pamiętająco-wysyłającym (ang. store-and-forward). W sieci lokalnej działa na poziomie transportowym lub dostępu do ośrodka (ang. medium access, MAC). Do podstawowych funkcji pomostów należy:

- zwiększenie odległości fizycznych w ramach praktycznych ograniczeń wynikających z dopuszczalnej szybkości i tłumienności,
- zwiększenie maksymalnej liczby węzłów,
- rozdzielenie lokalnych społeczności i regulowanie ruchu (tj. redukowanie przeciążeń).

Jest oczywiste, że pomost na poziomie dostępu do ośrodka będzie działał szybciej niż na poziomie transportowym. Największa szybkość osiągana obecnie na poziomie dostępu do ośrodka wynosi ok. 500 Kb/s, choć wkrótce może wynosić już 2 Mb/s. Pomosty lokalne spełniają głównie dwie ostatnie funkcje, natomiast pomosty odległe zapewniają spełnienie wszystkich.

Podstawowym problemem wymagającym rozwiązania przy tworzeniu pomostów jest trasowanie (ang. routing), co sprowadza się do znajomości — dla każdego pomostu — miejsc oznaczanych przez odpowiednie adresy. W tym celu każdy pomost utrzymuje swą lokalną bazę danych, zapewniając oczywiście niepowtarzalność poszczególnych adresów. Bardzo ważna jest również kwestia szybkiego przeglądania bazy danych.

Zwykła metoda trasowania przy wykorzystaniu bazy danych dla pomostu na poziomie dostępu do ośrodka polega na dynamicznym nadzorowaniu adresów ramki lub na kierowaniu ramki do wszystkich kanałów za wyjątkiem tego, z którego pochodzi. Tę drugą technikę nazywa się trasowaniem rozpływowym (ang. flooding). Ponadto, w celu zminimalizowania rozmiaru bazy danych i rozpoznawania fizycznych przemieszczeń stacji przechowuje się i aktualizuje wiek każdego adresu. Tak więc, oddzielna dla każdego pomostu baza danych ma charakter dynamiczny i adaptacyjny.

O wiele trudniejsze niż utrzymanie bazy danych trasowania jest podejmowanie decyzji co do wysyłania lub nie wysyłania ramki i unikanie przeciążeń. Można to wykonać metodą przeszukiwania, która dla N elementów wymaga czasu proporcjonalnego do logN, lecz częściej stosuje się tablice mieszające (ang. hash tables), redukując w ten sposób czas przeszukiwania dla kilku cykli procesora. Proponowano używanie do tego celu sprzętowej pamięci asocjacyjnej, lecz jak dotąd nie zrealizowano w ten sposób żadnego pomostu.

Oczywiście, podczas podejmowania decyzji co do trasowania lub ze względu na zajętość żądanego kanału, a także po wykryciu kolizji, odbierana ramka musi być zapamiętana. Dlatego w rzeczywistości każdy pomost zawiera bufory nadawcze i odbiorcze. Gdy bufory są pełne z jakiegoś powodu, wtedy występuje przeciążenie. Praktycznie dostępne pomosty mają na ogół bardzo duże bufory dla ramek i wystarczającą pamięć do przechowywania adresowej bazy danych (typowo — do 8000 adresów), tak że obecnie można osiągnąć przepustowość rzędu 3000 ramek na sekundę.

Jedna bardzo skuteczna technika rozszerzania obszaru objętego siecią Ethernet polega na użyciu odległych pomostów połączonych światłowodami, dzięki czemu można osiągnąć odległość 10—20 km. Innym rozwiązaniem jest użycie do tego celu systemu szerokopasmowego, gdzie pomosty są z jednej strony sprzężone z siecią Ethernet, a z drugiej — z urządzeniem szerokopasmowym (modemem). W jednej ze znanych realizacji takiego pomostu używa się trzech kanałów CATV o szerokości 6 MHz, przeznaczając 14 MHz na kanał danych o szybkości 10 Mb/s, a 2 MHz na wykrywanie kolizji.

Bramy

Problem sprzężenia sieci Ethernet z inną siecią, niekoniecznie siecią lokalną, rozwiązuje się stosując bramy. Zasadnicza różnica między pomostem a bramą nie polega tylko na tym, że ośrodki po obu stronach bramy są różne, lecz na koniecznej konwersji protokołów obowiązujących w obu sieciach połączonych bramą.

W najprostszym wypadku, sieć Ethernet można połączyć bramą z pierścieniem Token Ring IEEE 802.5, rozwiązując problem na poziomie dostępu do ośrodka, ponieważ protokoły wyższych warstw pozostają niezmienione. Do tego celu używa się w zasadzie zmodyfikowanego pomostu. W praktyce może to być połowa pomostu z jednostką pamiętająco-wysyłającą (ang. store and forward unit) odpowiednią dla pierścienia.

W bardziej złożonych wypadkach, bramy służą do sprzęgania sieci o całkowicie różnych charakterystykach i protokołach. Istnieje wiele takich bram do łączenia sieci Ethernet z innymi sieciami lokalnymi lub rozległymi, np. DECNET, SNA lub X.25, lecz także opartymi na protokołach wysokiego poziomu XNS lub ISO.

Oczywiście, funkcje bramy mogą być znacznie bardziej złożone niż pomostu. Przykładowo, brama do sieci X.25 powinna obsługiwać przychodzące i wychodzące wywołania, zapewniać dostęp do wywołań i autoryzację, przechowywać dane statystyczne itd. Tego rodzaju problemy rozwiązywano jednak wielokrotnie w typowych realizacjach mikroprocesorowych.

USŁUGI GLOBALNE I WYDAJNOŚĆ SIECI

Usługi globalne dla całej sieci są cechą charakterystyczną sieci lokalnej Ethernet, lecz wskutek braku normalizacji ich realizacje są znacznie zróżnicowane. Problem jest bardzo trudny do rozwiązania, zarówno pod względem technicznym jak i dlatego, że zwykle niełatwo zintegrować duże środowisko użytkowników i zasobów. Typowymi usługami globalnymi są: obsługa plików i poczta elektroniczna. O ile stacje obsługi plików dla sieci Ethernet są dostarczane przez kilku producentów, to perspektywy rozwiązania problemu poczty elektronicznej są mniej jasne, choć istnieje tendencja do realizowania usług pocztowych przy użyciu protokołu X.400. Należy oczekiwać, że w dłuższym okresie takie usługi jak poczta elektroniczna czy kontrola dostępu zostaną ostatecznie zrealizowane w sieciach lokalnych przez stacje obsługi o nazwach **mail** lub **logon**, które będą wykonywać niezbędną kontrolę dostępu, trasowanie wiadomości i obsługę nazw, wspomagane przez wszystkie inne zasoby sieci. Jest to najlepszy sposób zapewnienia globalnej przestrzeni nazw dla całej sieci i zintegrowania usług dla użytkowników.

Zarządzanie siecią i jej usługami jest również problemem częściowo rozwiązanym. Służą do tego oddzielne stacje obsługi, zwane zarządcami sieci, zwykle oparte na stanowiskach roboczych, wyposażonych w dodatkowe oprogramowanie.

Posiadając lokalny lub zdalny dostęp do sieci, zarządca sieci może na ogół wykonywać następujące funkcje:

- ładowanie skrośne (ang. downloading), umożliwiające zdalne załadowanie programów do wszystkich innych stacji obsługi w sieci,
- nadzorowanie i tworzenie śladu usług, polegające na rejestrowaniu wszystkich głównych zdarzeń w sieci, jak np. połączeń (skutecznych i nieskutecznych) i rozłączeń, ruchu w sieci itd.,
- usługi konfiguracyjne, umożliwiające administratorowi sieci określenie dopuszczalnych charakterystyk indywidualnych urządzeń i portów, co zapewnia dynamiczną rekonfiguralność,
- obsługę nazw, pozwalającą na definiowanie i pamiętanie nazw logicznych,
- rozgłaszanie wiadomości do wszystkich użytkowników lub ich grup,
- diagnostykę sieci, pozwalającą na wykrywanie, a jeśli możliwe, także na zdalne poprawianie defektów; wykonując transmisję na własny adres, zarządca sieci może — na przykład — wykryć pętlę, a jeśli trzeba, dokonać rekonfiguracji.

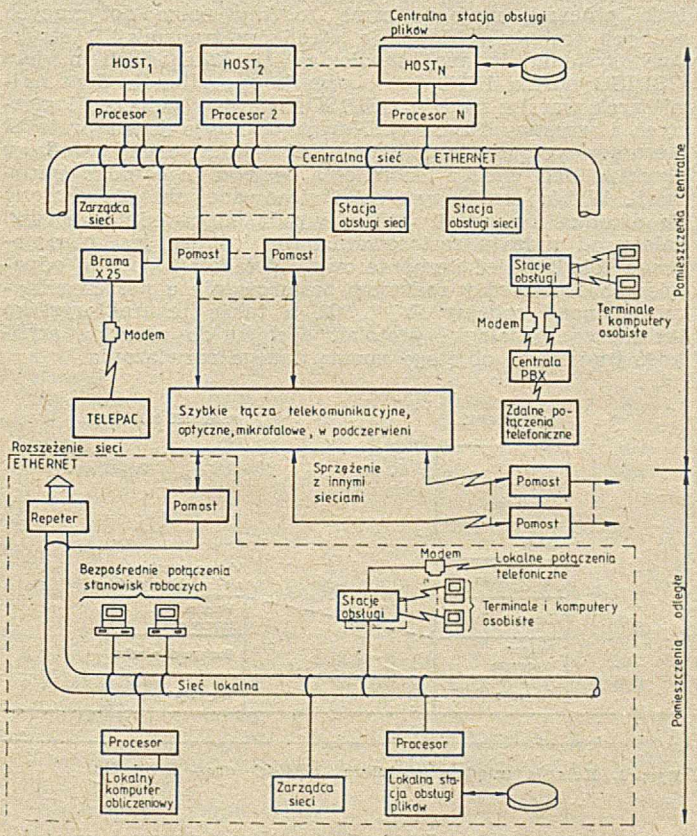
Obecnie jednostki nadzorowania sieci i zarządzania nią stały się nieodzowne w sieciach lokalnych wszystkich rodzajów. Należy jednak podkreślić, że funkcje zarządzania siecią mogą być zrealizowane tylko jako dodatkowe właściwości innych stacji obsługi sieci. Z tego powodu konieczne jest wprowadzenie normalizacji.

Prawie wszystkie istniejące obecnie realizacje sieci Ethernet są dość słabo obciążone, głównie dlatego, że używa się ich przede wszystkim do obsługi terminali. Jednakże, istnieje wystarczająco wiele dowodów na to, że w sieci Ethernet można osiągnąć szybkość transmisji danych rzędu 3 Mb/s. Nawet w bardzo obciążonej sieci dochodzi do niewielu kolizji. Jak stwierdzono, do kolizji dochodzi zaledwie w ok. 0,05 procenta wypadków, a tylko 1 procent wszystkich pakietów jest przesyłany z opóźnieniem. Tak więc, w przeciwieństwie do wielu obiegowych opinii, ruch w sieci Ethernet jest prawie deterministyczny.

Bardzo istotne, szczególnie dla obsługi plików, jest obciążenie procesora. Bardzo łatwo jest zużyć 50 procent czasu procesora, pracującego z szybkością 1 mips (ang. megainstruction per second, megainstrukcji na sekundę), jeśli służy on jako stacja obsługi plików w sieci Ethernet o dużym natężeniu ruchu i wykonuje większą część protokołu przesyłania plików. Jednakże, liczba użytkowników, którzy mogą być obsługiwani przez określoną stację obsługi plików sieci Ethernet, zależy od rodzaju zastosowania, dlatego konieczne jest przeprowadzenie dokładniejszych pomiarów w tym zakresie.

SIEĆ UNIWERSYTETU GENEWSKIEGO

Na Uniwersytecie Genewskim realizowany jest bardzo duży projekt utworzenia uniwersyteckiej sieci lokalnej, opartej na koncepcjach sieci Ethernet. Pełen schemat, przedstawiony na rys. 4, składa się z siedmiu sieci Ethernet sprzężonych w gwiazdę przez wspólną oś (ang. backbone). Do sprzężenia służą przede wszystkim odległe pomosty wykorzystujące światłowodowe łącza telekomunikacyjne. Każda sieć Ethernet ma jeden lub więcej komputerów macierzystych. Wszystkie terminale lub ich emulatory są sprzężone z siecią przez stacje obsługi. Połączenia są zrealizowane głównie przez grube kable i wielowejściowe repetery, lecz tworzy się także klastry (ang. clusters) komputerów osobistych sprzężonych przez sieć Cheapernet. W trakcie realizacji są bramy do sieci Telepac (X.25), EARN i innych sieci rozległych.



Rys. 4. Sieć komputerowa Uniwersytetu Genewskiego

Oś główna jest wykorzystywana nie tylko do sprzęgania z innymi sieciami, lecz także do łączenia centralnych komputerów macierzystych, do realizacji usług interakcyjnych, do rozwiązywania dużych problemów obliczeniowych oraz do realizacji usług administracyjnych i obsługi baz danych. Obsługa plików jest hierarchiczna i rozproszona; klastry współdzielą lokalne stacje obsługi, a każda sieć Ethernet ma także co najmniej jedną stację obsługi plików. Jednakże, archiwizacja i rezerwowanie plików (ang. file backup) są dokonywane centralnie, gdzie jest dostępna szybka pamięć o bezpośrednim dostępie i jednostki taśmowe (ang. streamer). Przewiduje się wprowadzenie trzech poziomów drukarek bezuderzeniowych, działających na zasadzie stacji obsługi — drukarki biurowe, drukarki zbiorcze i wysokiej jakości drukarki centralne.

Najtrudniejszy do rozwiązania jest problem zintegrowania usług. Obecnie planuje się wprowadzenie takich usług, jak: obsługa nazw, obsługa plików, poczta elektroniczna, skład dokumentów i wiele innych. Zainstalowanie wszystkich urządzeń jest przewidziane na połowę 1987 roku, lecz planuje się ciągły rozwój i unowocześnianie dołączanych komputerów osobistych i stanowisk roboczych. Sieć będzie obsługiwać ok. 3000 użytkowników.

* * *

Dotychczas rozwiązano wszystkie podstawowe problemy realizacji sieci Ethernet według specyfikacji IEEE 802.3 i produkuje się odpowiedni sprzęt. Przykładowo, możliwe jest rozmieszczenie sieci lokalnej na dużych obszarach geograficznych, o odległościach rzędu dziesiątków kilometrów i połączenie przez bramy z sieciami rozległymi i innymi rodzajami sieci. Bez trudności można też kupić sprzęt i oprogramowanie umożliwiające sprzężenie z siecią Ethernet terminali, stacji obsługi, komputerów macierzystych i innych urządzeń.

Najważniejszym problemem do rozwiązania w najbliższym okresie jest wybór protokołów wysokiego poziomu i zintegrowanie usług w sposób dogodny dla użytkowników. Na razie, coraz szerzej przyjmuje się protokół TPC/IP, lecz w użyciu jest wiele wzajemnie niezgodnych metod zarządzania siecią, kontroli dostępu i obsługi nazw. Szczególnie pożądane jest opracowanie obejmujących te zagadnienia norm ISO i IEEE.

Do rozwiązania pozostają jednak wciąż dwie kwestie: ograniczenie natężenia ruchu oraz zintegrowanie usług cyfrowych i akustycznych. Oba te problemy mogą być rozwiązane przez następną generację lokalnych sieci komputerowych, mających większą szybkość (co najmniej 100 Mb/s), używających metody CSMA/CD lub dostępu znamieniowego (ang. token passing).

Tłumaczył i opracował:

Janusz ZALEWSKI

Systemy wielomikroprocesorowe (I)

dokończenie ze str. 3

Żądania jednostki wykonawczej dotyczące pobrania argumentu rozkazu lub zapisania wyniku rozkazu mają wyższy priorytet niż zapełnianie wewnętrznej kolejki jednostki sprzęgającej. Oznacza to, że w wypadku pojawienia się takiego żądania, jednostka sprzęgająca po zakończeniu bieżącego cyklu maszynowego realizuje zgłoszone żądanie, a następnie powraca do zapełnienia swojej wewnętrznej kolejki. Cykle dostępu do magistrali są na ogół krótsze niż cykle wykonywania rozkazów przez jednostkę wykonawczą. Z tego względu stosunkowo często zdarza się, że wewnętrzna kolejka jednostki sprzęgającej jest pełna. Jednostka sprzęgająca odłącza się wówczas od magistrali, zwalniając ją dla innych mikroprocesorów. Ponowne dołączenie się jej do magistrali może nastąpić w dowolnym cyklu zegarowym, bez potrzeby dodatkowej synchronizacji.

Wewnętrzny podział mikroprocesora na dwie asynchroniczne jednostki, wykonawczą i sprzęgającą, z możliwością buforowania pobieranych wyprzedzająco rozkazów, bardzo istotnie redukuje wpływ oczekiwania dostępu do magistrali na efektywność pracy mikroprocesora, umożliwiając tym samym budowę efektywnych systemów wielomikroprocesorowych.

KRZYSZTOF ZIELIŃSKI
JADWIGA INDULSKA
ROMAN KRASOWSKI
Instytut Informatyki
Akademia Górniczo-Hutnicza
Kraków

Lokalna sieć komputerowa UMMLAN-2

Od połowy lat siedemdziesiątych notuje się bardzo szybki rozwój badań oraz zastosowań systemów komputerowych, zwanych lokalnymi sieciami komputerowymi (LSK). Mimo stosunkowo długiej już historii rozwoju tych systemów brak jest precyzyjnej ich definicji.

Zazwyczaj mówi się [1, 10], że LSK stanowi system komputerów rozmieszczonych na ograniczonym obszarze geograficznym (o promieniu mniejszym niż 5 km) i połączonych podsystemem transmisji o następujących cechach:

- tani ośrodek transmisji
- tanie układy sprzęgające (modemy, nadajniki, odbiornik) do ośrodka
- łatwa realizacja dołączenia do ośrodka
- duża szybkość transmisji (od 1 do 50 Mb/s)
- niezależność szybkości transmisji w ośrodku od szybkości pracy urządzeń dołączonych do sieci
- potencjalna możliwość komunikacji każdego urządzenia dołączonego do sieci
- rzadkie występowanie centralnego układu sterującego dostępem do ośrodka transmisji
- w większości wypadków słyszalność nadawanych wiadomości przez każde urządzenie dołączone do sieci.

Wymienione własności podsystemu transmisji LSK sprawiają, iż systemy te wnoszą zupełnie nowe możliwości w zakresie budowy oraz organizacji przetwarzania informacji w systemach informatycznych. W szczególności umożliwiają:

— efektywne wykorzystanie zasobów fizycznych i logicznych systemu informatycznego przez łatwość komunikacji między wszystkimi urządzeniami,

— łączenie urządzeń o różnych parametrach sprzętowych oraz protokołach komunikacji,

— łatwą rozbudowę systemu przez dołączenie nowych urządzeń,

— uzyskanie wysokiego stopnia integracji przetwarzania i przechowywania informacji, co wyraża się możliwością organizacji przetwarzania wielokomputerowego oraz budowy rozproszonych baz danych,

— zwiększenie niezawodności działania systemu informatycznego.

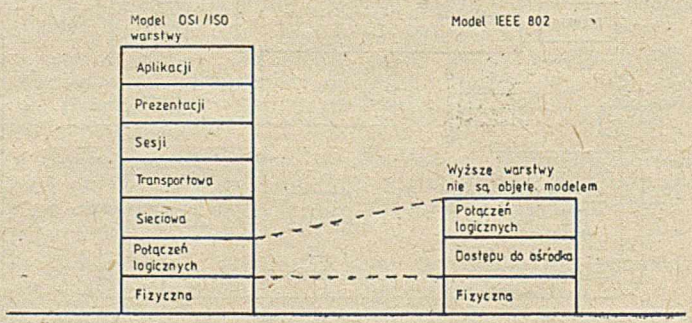
Wymienione czynniki zadecydowały, że LSK są powszechnie stosowane do budowy nowoczesnych systemów informatycznych i produkowane przez ponad sto różnych firm komputerowych.

Prace nad rozwojem LSK prowadzi się również w naszym kraju w kilku ośrodkach akademickich [4], między innymi od kilku lat w Instytucie Informatyki Akademii Górniczo-Hutniczej [2, 3, 4, 5].

W artykule przedstawiono aktualny stan prac standaryzacyjnych w odniesieniu do LSK, istniejące układy VLSI, stanowiące podstawę konstrukcji ich podsystemu transmisji, oraz omówiono na tym tle architekturę sprzętową i oprogramowanie sieci UMMLAN-2 (University of Mining and Metallurgy Local Area Network) zbudowanej w II AGH.

PRACE NORMALIZACYJNE I ROZWÓJ TECHNOLOGII VLSI DLA LSK

Zakres prac normalizacyjnych dotyczących LSK celowo jest rozważać w odniesieniu do ogólnego Modelu Referencyjnego OSI/ISO [6, 7, 9], zawierającego specyfikacje architektury sieci komputerowych. Model ten zawiera opis funkcjonalny siedmiu warstw, które zostały w tej architekturze wyróżnione. Model OSI/ISO jest obecnie powszechnie przyjęty i umożliwia ujednolicenie budowy sieci komputerowych. Równolegle do pracy nad modelem OSI/ISO RM (na początku lat osiemdziesiątych) były prowadzone prace komitetu IEEE 802 dotyczące normalizacji podsystemu transmisji dla lokalnych sieci komputerowych. Stąd pewne pojęcia oraz zasada podziału systemu na warstwy w modelu IEEE 802 LAN, stanowiącego wynik prac komitetu 802, nie są w pełni zgodne z modelem OSI/ISO. Porównanie zakresu i wzajemną relację pomiędzy modelami przedstawiono na rysunku 1. Wynikiem prac komitetu 802 jest nie tylko specyfikacja funkcjonalna, jaką stanowi model OSI/ISO, lecz również specyfikacja realizacji warstw objętych pracami, mająca formę rodziny norm dotyczących różnych technik dostępu do ośrodków transmisji i struktury systemu (zwanej topologią). W kategoriach modelu OSI/ISO norma IEEE 802 obejmuje specyfikacje dwu najniższych warstw modelu OSI/ISO: warstwy połączeń logicznych i warstwy fizycznej. Norma definiuje jeden protokół sterujący połączeniami logicznymi oraz cztery rodzaje struktur sieci wraz z technologią dostępu do ośrodka transmisji. Wybór struktury sieci i stosowanej metody dostępu do ośrodka transmisji (zatem wybór odpowiedniej normy) zależy od rodzaju zastosowania, dla którego jest przeznaczona lokalna sieć komputerowa. W tabeli 1 podano odpowiedniość pomiędzy rodzajem zastosowania a poszczególnymi normami rodziny IEEE 802, a także charakterystykę techniczną (topologię, protokół dostępu do ośrodka, szybkość transmisji) objętego normą danego rozwiązania.



Rys. 1. Relacja między modelem OSI/ISO oraz IEEE 802

Komputery, węzły lokalnej sieci komputerowej są dołączone do ośrodka transmisji (kabla koncentrycznego, światłowodu) przez urządzenia, sterowniki realizujące najniższe warstwy modelu OSI/ISO (zwykle dwie dolne warstwy modelu). Rozwój technologii LSI/VLSI umożliwia uproszczenie projektowania i oprogramowania tych urządzeń. Obecnie produkowane układy VLSI dla LSK realizują warstwę połączeń logicznych i zastępują od 60 do 100 układów SSI/

/MSI, a współpracujące z nimi układy szybkiej transmisji szeregowej pozwalają na eliminację od 10 do 16 układów SSI/MSI. Zmniejsza się też pobór mocy. Przewiduje się, że wprowadzone przez układy VLSI uproszczenia projektowania i oprogramowania sterowników lokalnych sieci komputerowych spowodują spadek o około 70% kosztu dołączenia komputerów do sieci LSK.

Tabela 1. Charakterystyka norm LSK

Norma	Charakterystyka techniczna	Preferowane zastosowania
IEEE 802.3	Topologia szynowa: protokół dostępu CSMA/CD, szybkość transmisji od 1 do 20 Mb/s	Systemy zarządzania, systemy automatyzacji biur, systemy czasu rzeczywistego niezbyt silnie uwarunkowane czasowo
IEEE 802.4	Topologia szynowa: protokół dostępu z przekazywaniem znacznika (ang. token passing), szybkość transmisji od 1 do 10 Mb/s	Systemy silnie uwarunkowane czasowo, systemy sterowania produkcją
IEEE 802.5	Topologia pierścieniowa: protokół dostępu z przekazywaniem znacznika (ang. token passing), szybkość transmisji od 1 do 10 Mb/s	Systemy silnie uwarunkowane czasowo, systemy sterowania produkcją
IEEE 802.6	Sieć regionalna (ang. metropolitan), norma budowy sieci miejskich obejmujących sieci LSK oraz elementy sieci zdalnych	Sieć miejska, rozszerzenie działania sieci lokalnej na większe obszary przez połączenie technik sieci lokalnych i zdalnych

Obserwowana tendencja w zastosowaniach VLSI w lokalnych sieciach komputerowych wskazuje, że w przyszłości również następne warstwy modelu OSI/ISO (zmodyfikowane dla potrzeb LSK) zostaną zrealizowane w postaci układów scalonych. Większość dostępnych obecnie na rynku światowym układów scalonych dla LSK pojawiła się w roku 1984. W tabeli 2 podano charakterystykę produkowanych układów VLSI wykorzystywanych do realizacji podsystemu transmisji LSK.

Tabela 2. Układy VLSI stosowane w realizacji podsystemu transmisji LSK

Firma	Oznaczenie układu	Protokół	Szybkość transmisji
AMD	AM7990 LANCE	CSMA/CD	10 Mb/s
Harris	Hd-15530-8	1553	2,5 Mb/s
Intel	82586	CSMA/CD	10 Mb/s
Mostek	MK68590 LANCE	CSMA/CD	10 Mb/s
Seeq	8001 EDLC	CSMA/CD	10 Mb/s
Standard Microsystems	COM9026	Token Passing	2,5 Mb/s
Ungermann-Bass			
Fujitsu	MB61301 DLC	CSMA/CD	10 Mb/s
Western Digital	WD2840	HDLC/Token Passing	1 Mb/s

ARCHITEKTURA SPRZĘTOWA SIECI UMMLAN-2

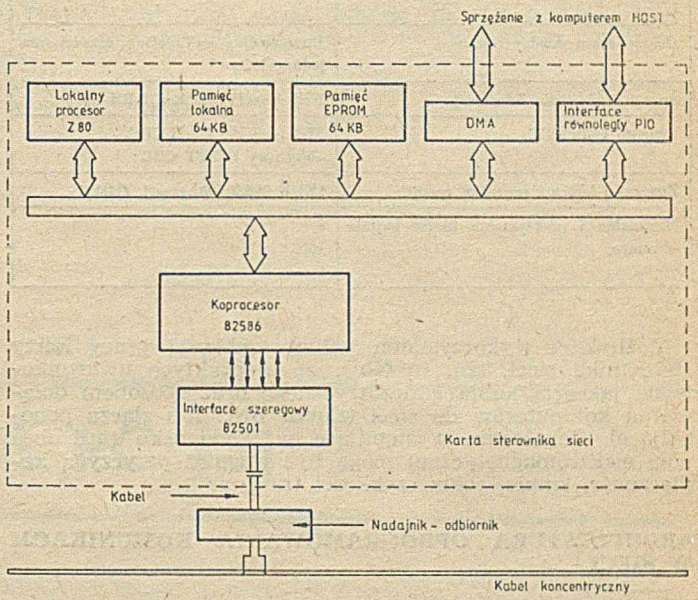
Architektura sprzętowa sieci UMMLAN-2 odpowiada założeniom normy IEEE 802.3. Zgodnie z tą normą sieć UMMLAN-2 ma następujące cechy charakterystyczne:

— topologia szynowa,
— kabel koncentryczny jako ośrodek transmisji,
— równorzędność dostępu do ośrodka wszystkich dołączonych urządzeń,
— protokół dostępu CSMA/CD.

Fizyczne dołączenie do sieci składa się z karty sterownika sieci, kabla nadajnika-odbiornika oraz nadajników i odbiorników linii. Na rysunku 2 przedstawiono schemat blokowy karty sterownika sieci.

Interfejs szeregowy 82501 realizuje następujące funkcje:

- kodowanie i dekodowanie informacji w kodzie Manchester (lub NRZ);
- wykrywanie ciągłej transmisji — funkcja polegająca na wykryciu sytuacji, w której uszkodzony sterownik nadaje w sposób ciągły informacje; w praktyce oznacza to przerwanie transmisji po 25 ms;
- wykrywanie kolizji — funkcja stanowiąca podstawowy element protokołu CSMA/CD polegająca na wykryciu sytuacji, w której dwa lub więcej sterowników równocześnie transmituje informacje przez kabel;
- wykrywanie nośnika informacji, czyli wykrywanie zajętości kabla dla transmisji, realizowane przez pewien sterownik sieci;
- filtracja zakłóceń występujących w kablu;
- odtwarzanie sygnału zegarowego dla sygnału zakodowanego;
- tworzenie pętli wewnętrznej (odbiór sygnału nadawanego przez układ 82501 przez jego układ odbiorczy), co umożliwia testowanie karty sterownika sieci niezależnie od układów odbiornika i nadajnika połączonych z kablem koncentrycznym.



Rys. 2. Architektura sterownika sieci UMMLAN-2

Koprocesor 82586 stanowi najbardziej zaawansowany pod względem technologicznym element karty sterownika i decyduje o jego dużej elastyczności i wysokich parametrach. Koprocesor ten realizuje cztery klasy funkcji:
1) nadawanie ramki informacji,
2) odbiór ramki informacji,
3) zarządzanie transmisją,
4) funkcje diagnostyczne.

Procesor Z80-CPU przygotowuje w pamięci komendy oraz bufory dla koprocesora 82586, a także wykonuje programy warstwy transportowej. Pamięć dynamiczna 64 KB służy do buforowania informacji nadawanej i odbieranej oraz do przechowywania programów realizujących warstwę transportową. Pamięć EPROM 16 KB zawiera kopię oprogramowania karty sterownika sieci. Układy DMA oraz PIO Z80 stanowią sprzężenie do magistrali komputera włączonego do sieci.

Proponowane rozwiązanie karty sterownika ma następujące cechy:
- niewielką liczbę elementów, co powinno zapewnić dużą niezawodność
- możliwość umieszczenia całości oprogramowania warstwy transportowej oraz sprzężenia do warstwy połączeń logicznych na karcie sterownika w pamięci EPROM, co umożliwia duże uproszczenie, zwłaszcza pod względem programowym, łączenia w sieć różnych komputerów
- dużą uniwersalność sprzężenia z magistralą komputera włączonego do sieci, co umożliwia łatwe zastosowanie karty w różnych typach komputerów
- łatwość dostosowania się do istniejącego w sieci (np. już posiadanej przez użytkownika) formatu wiadomości oraz parametrów czasowych; wynika to z faktu zastosowania koprocesora 82586 umożliwiającego transmisję z szybkością do 10 Mb/s oraz ustawienie w sposób programowy wielu parametrów (możliwość skonfigurowania układu przedstawiono w tabeli 3).

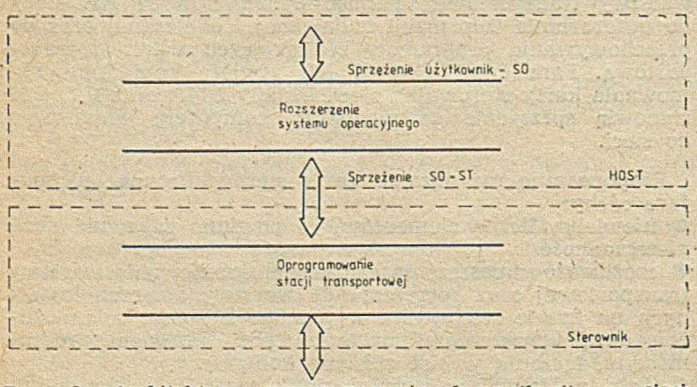
Tabela 3. Możliwości skonfigurowania koprocesora 82586 ze względu na format ramki i parametry czasowe

Parametr	Możliwe warianty
Długość preambuły (ang. preamble)	2 bajty 4 bajty 8 bajtów 16 bajtów
Długość adresu	od 0 do 6 bajtów
Priorytet liniowy	6 bitów
Priorytet eksponencjalny (przyśpieszone rozstrzyganie konfliktu)	2 bity
Metoda naliczania opóźnienia transmisji po wystąpieniu kolizji	IEEE 802.3 (Ethernet)
Długość szczeliny czasowej stanowiącej jednostkę opóźnienia transmisji po wystąpieniu kolizji	Liczba długości 2 bajty × czas trwania jednego bitu
Odstęp czasu między ramkami	Liczba o długości 7 bitów × czas trwania jednego bitu
Sposób kodowania informacji	Kod Manchester, kod NRZ
Długość kodu CRC	32-bitowy Atodin II CRC, 16-bitowy CCITT CRC
Znaczniki końca i początku ramki	IEEE 802.3 (Ethernet), HDLC
Minimalna i maksymalna liczba bajtów w ramce	8 1508

Możliwości wykorzystania pełnej szybkości pracy karty sterownika sieci, tzn. 10 Kb/s, są w praktyce uwarunkowane jakością kabla koncentrycznego oraz sposobem dołączenia komputerów do sieci (złącza BNC lub złącza penetrujące). Zakłócenia występujące w środowisku (np. obce pola elektromagnetyczne) mogą być również przyczyną konieczności zmniejszenia szybkości transmisji.

ARCHITEKTURA OPROGRAMOWANIA KOMUNIKACJI W SIECI

Przy projektowaniu oprogramowania sieci występuje zawsze problem podziału siedmiu warstw oprogramowania (według modelu OSI/ISO) między sterownik i komputer Host (węzeł sieci). W sieci UMMLAN-2 wprowadzono następujący podział funkcji: sterownik realizuje warstwy 1—4 modelu (warstwy ukierunkowane na komunikację wraz z warstwą transportową), a pozostałe warstwy są realizowane w komputerze Host jako rozszerzenie systemu operacyjnego (rys. 3). Taki podział funkcji wynika z przyjętych założeń dotyczących konstrukcji uniwersalnego podsystemu transmisji, a przyjęta dla sterownika architektura sprzętowa umożliwia pełną realizację czterech dolnych warstw modelu OSI/ISO.



Rys. 3. Architektura oprogramowania komunikacji w sieci UMMLAN-2

Sterownik sieci UMMLAN-2 stanowi dla komputera Host stację transportową i realizuje przesyłanie wiadomości przekazywanych z tego komputera. Wiadomość może być przesyłana w trybie połączeniowym (bez zagubień, bez duplikatów i z zachowaniem sekwencji) lub w trybie bezpołączeniowym (ang. connectionless) nie gwarantującym niezawodnego przesłania (tzw. datagram). Działaniem stacji transportowej steruje komputer Host za pomocą zbioru komend, stanowiących sprzężenie między stacją transportową a systemem operacyjnym tego komputera (sprzężenie ST-SO). Zbiór ten, oprócz podstawowych komend wysyłania i odbioru wiadomości, zawiera również komendy inicjujące działanie stacji transportowej oraz komendy umożliwiające realizację trybu połączeniowego (operacje na portach). Komendy sprzężenia ST-SO przedstawiono w tabeli 4.

Tabela 4. Zbiór komend sprzężenia SO-ST w sieci UMMLAN-2

Komenda	Semantyka komendy
Init	Inicjalizacja stacji transportowej
Address	Nadawanie węzłowi sieci adresu
Configure	Definicja (lub redefinicja) parametrów stacji transportowej i koprocesora 82586
Diagnose	Diagnostyka stacji transportowej i koprocesora
Start	Początek trybu przesyłania i odbioru wiadomości w sieci
Create _ Port	Tworzenie portu
Destroy _ Port	Likwidacja portu
Test _ Port	Testowanie stanu portu
Send	Przesłanie wiadomości
Receive	Odbiór wiadomości
Preview	Testowanie portu na przybycie wiadomości

Na obecnym etapie konstrukcji sieciowego systemu komputerowego, system operacyjny został rozszerzony o komunikację międzyprocesową, umożliwiającą wymianę wiadomości między programami użytkownika rozproszonymi w sieci oraz między programami użytkownika a procesami systemu operacyjnego obsługującymi zdalny dostęp do zasobów sieci. Procesy systemu operacyjnego sieci, umożliwiające zdalny dostęp do zasobów, są przedmiotem aktualnie prowadzonych prac.

Do realizacji komunikacji międzyprocesowej system operacyjny udostępnia programom użytkowym zbiór komend tworzących tak zwane sprzężenie **Użytkownik-SO.** Do zbioru komend tego sprzężenia należą komendy przesłania i odbioru wiadomości oraz komendy inicjujące działanie stacji transportowej, komendy diagnostyczne i operacyjne na portach, umożliwiające realizację niezawodnego przesyłania wiadomości w sieci (tab. 5). Do obsługi tych komend system operacyjny wykorzystuje własne procedury obsługi oraz zbiór komend sprzężenia **SO-ST.**

Tabela 5. Zbiór komed sprzężenia Użytkownik-SO w sieci UMMLAN-2

Komenda	Semantyka komendy
INSTALL	Inicjalizacja sterownika
DIAGNOSE	Diagnostyka sterownika
ALLOCATE	Tworzenie portu
DEALLOCATE	Likwidacja portu
TEST	Testowanie stanu portu
SEND	Przesłanie wiadomości
RECEIVE	Odbiór wiadomości
PREVIEW	Testowanie portu na przybycie wiadomości
WAITS	Zawieszenie procesu do czasu zakończenia operacji zleconej stacji transportowej
TESTOP	Testowanie zakończenia operacji zleconej stacji transportowej

Systemem operacyjnym, który jako pierwszy rozszerzono o zbiór komend umożliwiających komunikację międzyprocesową, jest system operacyjny CP/M dla mikrokompute-

rów 8-bitowych. Obecnie prowadzone są prace nad realizacją sprzężenia **Użytkownik-SO** dla systemów operacyjnych mikrokomputerów 16-bitowych.

* * *

Aktualne doświadczenia w dziedzinie konstrukcji LSK wskazują, że realne możliwości budowy tych systemów w Polsce są uwarunkowane dostępem do odpowiednich układów VLSI. Implikuje on bowiem zmniejszenie nakładu pracy związanego z konstrukcją sprzętu, eliminuje konieczność oprogramowania dwóch dolnych warstw modelu OSI/ISO oraz zwiększa niezawodność i parametry komunikacji w sieci.

Dla zmniejszenia nakładów na oprogramowanie dołączenia kolejnych rodzajów komputerów do sieci, celowa jest realizacja warstw 1—4 modelu OSI/ISO poza łączonymi komputerami. Natomiast na obecnym etapie rozwoju sieci lokalnych w Polsce realizacja warstw wyższych może być wykonana przez modyfikację istniejących systemów operacyjnych łączonych komputerów.

LITERATURA

[1] Gee K. C. E.: Local Area Networks, NCC Publications, Manchester, 1982

[2] Gościński A., Zieliński K.: The Design of Distributed Control Computer Systems. Computers in Industry, Vol. 6, pp. 37—45, 1985

[3] Gościński A., Zieliński K.: Mikrokomputerowa sieć lokalna do sterowania. Materiały Ogólnopolskiej Konferencji Naukowo-Technicznej „Mikrokomputery w automatyce i technice systemów", cz. III, Wrocław 1984

[4] Gościński A., Zieliński K.: Wykorzystanie sieci komputerowej do łączenia środków komputerowych w Instytucie Informatyki AGH. Prace IV Krajowej Konferencji Naukowo-Technicznej „Zastosowanie komputerów w przemyśle", str. 260—270, Szczecin, 1985

[5] Gościński A., Zieliński K.: Microprocessor Based UMM Local Area Network, Proc. of the Intern. School on Microcomputers and their Application, Karpacz, March 1984

[6] Harrison F. J.: IEEE Project 802 — Local Area Network Standard. Proc. Conf. Distributed Computer Control Systems, Gelle R. W. (ed.), Pergamon Press, Oxford, 1982

[7] IEEE 802 Local Network Standard — Draft B. IEEE, New York, 19 October 1981

[8] Kryskow I. M., Miller C. M.: Local Area Network Overview — Part 1 and 2. Computer Design, February and March 1981

[9] The Ethernet Local Area Network. Data Link Layer and Physical Layer Specifications, Version 1.0, DEC, INTEL, XEROX, 1980

[10] The LOCAL Netter Design Handbook. 2nd Edition, Architecture Technology Corporation, March 1983.

WALDEMAR DWORAKOWSKI
Instytut Maszyn Matematycznych
Warszawa

Komputery osobiste IBM PC (3)

W bieżącej części omówiono kolejne sterowniki urządzeń zewnętrznych komputera osobistego IBM PC dostępne i popularne na rynku polskim.

Sterownik monochromatycznego monitora graficznego

Obecnie bardzo rozpowszechnioną płytką jest monochromatyczny sterownik alfanumeryczny z możliwościami pracy w trybie graficznym — typu Herkules II. Płytka ta jest w zasadzie kompatybilna z monochromatycznym sterownikiem alfanumerycznym firmy IBM. Posiada ona jednak dodatkowo tryb graficzny o rozdzielczości poziomej 720 punktów i pionowej 348 punktów (dla porównania — płytka grafiki kolorowej umożliwia uzyskanie rozdzielczości 640 × 200 punktów). O ile ta płytka nie jest zainstalowana łącznie z płytą grafiki kolorowej, to możliwa jest praca na dwóch stronach grafiki, co umożliwia uzyskanie animacji obrazu. Niestety, ten tryb pracy nie może być wykorzystywany, jeżeli w komputerze jest umieszczona płyta grafiki kolorowej. System operacyjny zapewnia oprogramowanie alfanumeryczne i graficzne dla płyty grafiki kolorowej, jednakże dla opisywanej płyty zapewnione jest jedynie oprogramowanie trybu znakowego. Należy jednak dodać, że istnieje dość ograniczony zestaw programów współpracujących z tą płytą w trybie graficznym.

Płytka prototypowa

Płytka prototypowa umożliwia opracowywanie nowych kart, które mają współpracować z systemem IBM PC. Zawiera układy dekodowania adresu dla przestrzeni we-wy, układy bufora adresowego, bufora szyn sterujących oraz bufora szyny danych. Wbudowane układy dekodowania wytwarzają aktywny niski poziom sygnału, gdy adres rozkazu we-wy pochodzi z zakresu 300H-31FH. Płytka buforuje niektóre sygnały sterujące i linie adresowe oraz wszystkie linie danych, ułatwiając konstruowanie przez użytkownika i montaż układów prototypowych, które mają współpracować z mikrokomputerem. Płytki o podobnej funkcji, lecz innej konstrukcji, są obecnie produkowane przez różne firmy w kraju, m.in. COMPUTEX. Płytka ta, choć funkcjonalnie zgodna z płytką oryginalną, jest jednak oparta na innych elementach i zawiera bardziej elastyczny układ dekodowania.

Karta ekstendera i blok rozbudowy

Mikrokomputer IBM PC jest systemem stosunkowo łatwo rozszerzalnym i jeżeli użytkownik stosuje jedynie moduły o stosunkowo niewielu funkcjach na pojedynczym pakiecie, to szybko może dojść do sytuacji zapełnienia wszystkich miejsc na płycie głównej. W związku z tym firma IBM opracowała tzw. kartę ekstendera (ang. Extender Card) i blok rozbudowy (ang. Expansion Box), które łącznie umożliwiają ograniczone rozszerzenie konfiguracji komputera. Płytka ekstendera jest dołączana do złącza J8 komputera i umożliwia dołączenie jednego bloku rozbudowy. Blok rozbudowy zawiera zasilacz (65, 135 lub 170 W) oraz płytę główną (ang. Motherboard), z pięcioma lub ośmioma złączami, umocowane w obudowie takiej jak komputer główny. Istnieją różne ograniczenia, jeżeli chodzi o rodzaje płytek, które mogą być wykorzystywane w bloku rozbudowy. Wynikają one zarówno ze względów praktycznych jak i ze względu na konstrukcję karty ekstendera. Karta ta zawiera nie tylko układy buforowania, lecz także wiele dodatkowych układów wymaganych przez oprogramowanie systemowe lub przeznaczonych do testowania. W związku z powstawaniem płytek integrujących coraz więcej funkcji, użyteczność omawianego zestawu staje się coraz bardziej problematyczna, szczególnie jeśli się widzi jego wady i ograniczenia. Firma IBM stosuje omawiany zestaw w swoich mikrokomputerach specjalizowanych, wykorzystujących jako jednostkę centralną IBM PC.

Adaptery protokołów BSC i SDLC

Karta sterownika komunikacyjnego dla protokołu BSC (ang. Binary Synchronous Communication) została opracowana przez firmę IBM i służy do komunikacji według tego protokołu. Została skonstruowana na podstawie mikroukładu 8251A i zawiera oprócz niego układ dekodera adresowego, bufor danych, porty równoległe służące do testowania oraz do sterowania liniami RS232, układ czasowy (służący do wytwarzania sygnałów uzależnień czasowych oraz

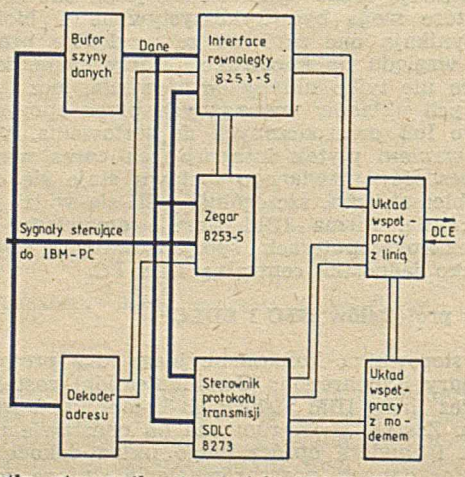
lokalnego sygnału zegara i do testowania) oraz układy konwerterów poziomów dla łączy RS232. W jednym komputerze mogą występować jedynie dwa układy tego rodzaju, przy czym stosowanie tej karty jest rozłączne ze stosowaniem pakietu interface'u komunikacyjnego SDLC/HDLC. Nie opracowano dotąd zbyt wiele oprogramowania współdziałającego z tym sprzętem.

Podobnie karta sterownika komunikacyjnego dla protokołu SDLC/HDLC (ang. Synchronous Data Link Control, Highlevel Data Link Control) została opracowana przez firmę IBM w celu komunikacji według tego protokołu. Została skonstruowana przy wykorzystaniu mikroukładu 8273 i zawiera oprócz niego podobne układy jak karta sterownika BSC: układ dekodera adresowego, bufor danych, porty równoległe, układ czasowy oraz układy konwerterów poziomów. W jednym komputerze mogą występować jedynie dwie karty tego rodzaju, przy czym nie stosuje się ich łącznie z kartą interface'u komunikacyjnego BSC. Oprogramowania przeznaczonego dla tego sprzętu jest również niewiele.

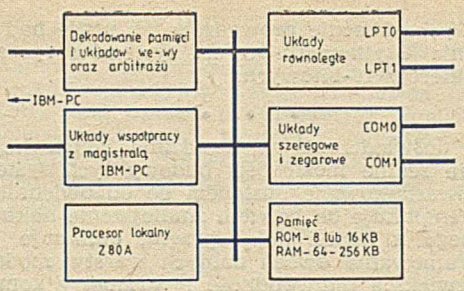
Procesor komunikacyjny CS88-Z80

Komputer IBM PC ma wiele wad, a jedną z nich jest oparcie całej konstrukcji na mocy obliczeniowej jednego procesora. Ponadto sprzedawany wraz z komputerem system operacyjny PC-DOS i sprzęt dodatkowy ogranicza liczbę standardowych urządzeń we-wy praktycznie tylko do dwóch drukarek i dwóch kanałów szeregowych. Pomimo że PC-DOS może obsłużyć więcej urządzeń (m.in. określone przez użytkownika), na przeszkodzie stają ograniczenia sprzętowe. Rozbudowę mikrokomputera utrudniają dodatkowe przeszkody, takie jak niewystarczająca pojemność pamięci w komputerze i ograniczona moc obliczeniowa procesora. Jednakże pojawiają się już na rynku płytki zawierające zarówno dodatkowe procesory o zastosowaniach obliczeniowych jak i procesory ukierunkowane na operacje wejścia-wyjścia. Jedną z takich płyt jest pakiet komunikacyjny CS88-Z80.

Zawiera on dwa kanały równoległe, dostosowane do współpracy z drukarkami i ploterami mającymi interface równoległy Centronics (Logabax lub IRPR — po zmianie oprogramowania) oraz dwa kanały szeregowe, które mogą pracować jako kanały asynchroniczne, synchroniczne lub w protokole HDLC. Pakiet ten zawiera ponadto pamięć RAM o pojemności do 256 KB, pamięć ROM z oprogramowaniem systemowym oraz procesor Z80A działający współbieżnie z procesorem znajdującym się na płycie głównej. Pamięć RAM nie koliduje z pamięcią systemu IBM PC, chociaż jest dostępna dla procesora głównego. Układy dostępu umożliwiają realizowanie operacji współbieżnych i ich synchronizację przez semafory, kolejki i system komunikatów. Wbudowane oprogramowanie obejmuje m.in. system operacyjny zgodny z CP/M 2.2 z wbudowaną obsługą dysków elastycznych, dysku Winchester, dysku RAM opartego na pamięci własnej lub pamięci płyty głównej, dysku ROM zawierającego podstawowe oprogramowanie systemowe, łączy komunikacyjnych, komunikacji z systemem PC-DOS itd. Obsługa protokołów transmisyjnych i komunikacja z systemem PC-DOS odbywa się przez oprogramowanie zapisane w pamięci ROM i na podstawie sterowników programowych.



Rys. 1. Płytka sterownika transmisji SDLC (DCE — ang. data communication equipment)



Rys. 2. Płytka procesora komunikacyjnego CS88-Z80

Oprócz zastosowania pakietu jako procesora komunikacyjnego dla komputera IBM PC, może on także służyć jako dodatkowy współbieżnie działający procesor obliczeniowy lub jako wspomagający system uruchomieniowy do programowania procesorów zgodnych z procesorami 8080 i Z80. Zainstalowane oprogramowanie, zgodne z CP/M 2.2, mogące działać współbieżnie z systemami PC-DOS lub Concurrent DOS (Concurrent CP/M) umożliwia wykorzystanie bogatej biblioteki oprogramowania dla systemu CP/M przy równoczesnym wykorzystaniu oprogramowania dostępnego w systemie PC-DOS. Omawiana płytka jest jedną z wielu oryginalnych konstrukcji firmy COMPUTEX.

Płytka sterownika sieciowego

W zastosowaniach komputerów typu IBM PC coraz powszechniejsze staje się wykorzystanie sieci lokalnych do tworzenia zestawów wielomaszynowych. Jednym z krajowych rozwiązań jest płytka sterownika sieciowego Cx-NET firmy Computex zgodna z siecią 10-NET, która wyróżnia się elegancją konstrukcji oraz dobrymi parametrami eksploatacyjnymi.

Płytka Cx-NET umożliwia tworzenie konfiguracji wielokomputerowych opartych na jednym kablu (skrętką 110-omowa), przy wykorzystaniu metody detekcji kolizji. Sygnał jest kodowany systemem Manchester, a poprawność transmisji jest zapewniana przy użyciu 32-bitowych wielomianowych kodów kontrolnych w 1024-bitowym pakiecie informacyjnym. Dzięki temu możliwe jest uzyskanie praktycznie 100% pewności wykrycia błędu transmisyjnego. Formowania pakietu dokonuje specjalizowany układ transmisyjny, a sama transmisja danych odbywa się z wykorzystaniem układów DMA, co powoduje odciążenie procesora. Płyta zawiera układy dekodera, układy buforujące szynę danych, interface sieci lokalnej Ethernet 8002, koder i dekoder kodu Manchester.

Do sterownika opracowano też bogate oprogramowanie umożliwiające współpracę wielu komputerów, działające w systemie DOS, wersja 3.10. Dodatkową zaletą tego rozwiązania jest istnienie oprogramowania wysokiego poziomu (np. bazy danych), wykorzystującego sieć i rozproszone struktury danych. Płytka Cx-NET może pracować z szybkością 1 Mb/s, co przy maksymalnej liczbie 32 komputerów w jednym segmencie sieci oznacza praktycznie brak kolizji. W wypadku konieczności połączenia większej liczby komputerów lub większych odległości między komputerami, możliwe jest wykorzystanie retransmiterów.

Oprócz komputerów, do sieci mogą być też dołączane tzw. terminale sieciowe, stanowiące komputery pozbawione pamięci masowych. Umożliwia to tworzenie konfiguracji opartych na niewielkiej liczbie komputerów wyposażonych w pamięci masowe o dużych pojemnościach i dużej liczbie tańszych komputerów o mniejszych możliwościach i pozbawionych pamięci masowych.

* * *

Artykuł ten stanowi jedynie pobieżny przegląd niektórych zagadnień i z oczywistych przyczyn nie może być kompendium wiedzy o układach towarzyszących dla komputerów typu IBM PC. Dokładne informacje o konstrukcji i parametrach sterowników można znaleźć w czasopismach fachowych lub w dokumentacjach technicznych tych urządzeń. Niestety, nie zawsze dokumentacja dostarczana przez producenta zawiera odpowiednie informacje, co mogło też mieć wpływ na dokładność informacji podanych w tym opracowaniu. Wobec niedokładności danych zawartych w literaturze firmowej i w związku z trudnościami w dotarciu do tej i innej literatury, wkrótce rozpoczniemy publikowanie artykułów koncentrujących się na szczególnie istotnych problemach konstrukcji sprzętu i oprogramowania dla systemów IBM PC.

ROMAN GRABOWICZ
JANUSZ ZALEWSKI
Warszawa

System operacyjny PC-DOS (3)

W bieżącej części opisu kończymy omawianie poleceń systemu operacyjnego.

POLECENIA ZWIĄZANE Z DYSKIEM STAŁYM

FDISK

Polecenie to umożliwia przydzielenie części lub całości dysku stałego dla systemu operacyjnego. Wszystkie operacje wykonywane na dysku stałym w systemie DOS muszą być poprzedzone założeniem, za pomocą polecenia **FDISK**, tzw. strefy systemu DOS (ang. partition). Polecenie **FDISK** nie może założyć ani usunąć stref przydzielonych innym systemom operacyjnym.

Rozpoczęcie wykonywania polecenia **FDISK** powoduje pojawienie się na ekranie menu, które pozwala użytkownikowi na wybór jednej z następujących operacji:

- utworzenie strefy systemu DOS
- zmiana strefy aktywnej
- usunięcie strefy systemu DOS
- wyświetlenie informacji o podziale dysku na strefy
- przejście do operacji na następnym dysku stałym.

Jeżeli na dysku nie ma podziału na strefy, to użytkownik może przydzielić cały dysk strefie systemu DOS. Strefie systemu DOS można przydzielić część dysku obejmującą spójny fragment, składający się z całkowitej liczby cylindrów. W tym celu podaje się rozmiar strefy w cylindrach oraz numer cylindra początkowego. Aby korzystać z dysku stałego, należy sformatować strefę poleceniem **FORMAT**. Jeżeli system operacyjny ma ładować się z dysku stałego, a nie z dyskietki w stacji **A**, to po założeniu strefy systemu DOS należy wykonać dodatkowe czynności:

1) wykonać polecenie FORMAT z parametrem **/S**
2) skopiować do tej strefy polecenia zewnętrzne systemu DOS
3) uczynić strefę systemu DOS aktywną (polecenie **FDISK** — operacja 2)
4) wyjąć dyskietkę systemową ze stacji **A**
5) przeładować system naciskając równocześnie klawisze **CTRL, ALT, DEL.**

Na dysku stałym mogą znajdować się cztery strefy, z których każda należy do innego systemu operacyjnego. Strefą aktywną jest ta, z której ładuje się system operacyjny. Tylko jedna strefa może być aktywna. Polecenie **FDISK** (operacja 2) umożliwia zmianę strefy z nieaktywnej na aktywną i odwrotnie. Jest to polecenie zewnętrzne.

BACKUP d:[path][filename[.ext]]
d:[/S][/M][/A]/[D:mm-dd-yy]

Jest to polecenie zewnętrzne, które powoduje składowanie plików podanych jako pierwszy parametr, znajdujących się na dysku stałym, na dyskietkę podaną jako drugi parametr. Dopuszcza się stosowanie zmiennych nazw i rozszerzeń plików do składowania (tj. znaków * i ?). Wtedy wszystkie odpowiednie pliki są składowane. Użytkownik może wykorzystać następujące parametry:

S — składowane są wszystkie pliki z podanego skorowidza i wszystkich jego podskorowidzów
M — składowane są tylko pliki, które zostały zmodyfikowane od poprzedniego składowania
A — składowane pliki są dopisywane do plików już istniejących na dyskietce; brak tego parametru (wartość domyślna) powoduje wymazanie plików już zapisanych
D — składowane są pliki zapisane nie wcześniej niż przed wyspecyfikowaną datą.

RESTORE d: [d:][path][filename][.ext]][/S][/P]

Jest to polecenie zewnętrzne, które odtwarza pliki z dyskietki znajdującej się w stacji, podanej jako pierwszy parametr, i zapisuje je na dysku stałym. Drugi parametr określa, które pliki i z którego skorowidza mają być odtwarzane. Pliki muszą być zeskładowane poleceniem **BACKUP**. Znaczenie parametrów jest następujące:

S — polecenie odtwarza wszystkie pliki z określonego skorowidza i wszystkich jego podskorowidzów

P — polecenie odtwarza pliki zmodyfikowane od ostatniego składowania i pliki o atrybucie Read-Only (tylko odczyt).

POLECENIA OPEROWANIA SKOROWIDZAMI

DIR [d:][path][filename][.ext][/P][/W]

Polecenie to powoduje wyświetlenie pozycji (nazw plików i skorowidzów) znajdujących się w określonym skorowidzu wraz z dodatkowymi informacjami, takimi jak rozmiar poszczególnych plików, wielkość wolnego obszaru na dysku itd. Polecenie dopuszcza użycie nazw zmiennych. Jeżeli podana zostanie nazwa pliku lub rozszerzenie nazwy, to wyświetlone zostaną tylko pozycje odpowiadające tej nazwie (i rozszerzeniu). Znaczenie parametrów jest następujące:

P — wyświetlanie jest zawieszane po każdym zapełnieniu ekranu
W — wyświetlane są tylko najważniejsze informacje w bardziej zwięzłej postaci.
Jest to polecenie wewnętrzne.

MKDIR [d:]path lub **MD** [d:]path

Jest to polecenie wewnętrzne, które zakłada nowy skorowidz określony przez ścieżkę dostępu. Oba formaty są równoważne.

CHDIR [[d:]path] lub **CD** [[d:]path]

Jest to polecenie wewnętrzne, które powoduje, że skorowidz określony jako parametr staje się skorowidzem bieżącym. Polecenie bez parametru lub z parametrem **d:** wyświetla ścieżkę dostępu do skorowidza bieżącego na wyspecyfikowanym lub domyślnym dysku.

RMDIR [d:]path lub **RD** [d:]path

Jest to polecenie wewnętrzne, które usuwa wyspecyfikowany skorowidz. Skorowidz nie może zawierać żadnych pozycji oprócz „." i „..". Nie można usuwać skorowidza bieżącego i systemowego.

TREE [d:][/F]

Polecenie to wyświetla wszystkie ścieżki dostępu znalezione na dysku określonej stacji, a także (opcjonalnie, jeśli użyty zostanie parametr **/F** nazwy i rozszerzenia nazw wszystkich plików w każdym skorowidzu. Jest to polecenie zewnętrzne.

Mgr ROMAN GRABOWICZ ukończył w 1979 r. Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, broniąc pracę dyplomową z zakresu systemów operacyjnych. Od 1979 roku pracuje w Wydziale Informatyki Polskich Linii Lotniczych LOT. Specjalizuje się w zastosowaniach mikrokomputerów w systemach zarządzania.

Zdjęcie i życiorys dr inż. JANUSZA ZALEWSKIEGO opublikowaliśmy w numerze 11—12/1981

PATH [[d:]path[[;[d:]path]...]]

Jest to polecenie wewnętrzne, które specyfikuje listę skorowidzów przeszukiwanych w celu ściągnięcia do pamięci operacyjnej poleceń zewnętrznych lub plików z poleceniami wsadowymi. Przeszukiwanie zawsze rozpoczyna się od skorowidza bieżącego, który nie musi być podany na liście parametrów polecenia **PATH**. Następnie przeszukuje się kolejne wyspecyfikowane skorowidze. Polecenie **PATH** bez parametrów wyświetla ostatnio wyspecyfikowaną listę skorowidzów, a polecenie **PATH** przywraca wartość domyślną, tzn. przeszukiwany jest tylko skorowidz bieżący.

POLECENIA FILTROWANIA

Filtrami nazywa się programy, które wczytują dane wejściowe ze standardowego urządzenia wejściowego, dokonują pewnych operacji na danych i zapisują przekształcone dane na standardowe urządzenie wyjściowe (zwykle ekran). Filtry stają się szczególnie użyteczne w połączeniu z przeadresowywaniem (ang. redirection). W systemie DOS istnieją standardowo trzy tego rodzaju polecenia zewnętrzne: **FIND, MORE** i **SORT**. Wszystkie trzy opracowano z uwagi na przetwarzanie plików tekstowych.

FIND [/V][/C][N] string[[d:]path] filename [.ext]...]

Jest to polecenie zewnętrzne, które wyszukuje w wyspecyfikowanym pliku (lub w plikach) podany ciąg znaków. Wiersze tekstu zawierające ten ciąg znaków są wysyłane na standardowe urządzenie wyjściowe. Znaczenie parametrów:

N — wyświetlany wiersz jest poprzedzony numerem określającym pozycję tego wiersza w pliku

C — wyświetla się tylko liczbę wystąpień ciągu znaków w pliku bez wyświetlania wierszy

X — wyświetla się tylko wiersze, które nie zawierają zadanego ciągu znaków.

MORE

Polecenie to wczytuje dane ze standardowego urządzenia wejściowego i przesyła zawartość pełnego ekranu na standardowe urządzenie wyjściowe. Następnie polecenie zawiesza się i oczekuje na naciśnięcie dowolnego klawisza w celu wyświetlenia kolejnego ekranu danych itd.

SORT [/R][/+n]

Polecenie to wczytuje dane (wiersze tekstu) ze standardowego urządzenia wejściowego, sortuje je i zapisuje posortowane dane (wiersze tekstu) na standardowe urządzenie wyjściowe. Sortowanie odbywa się według porządku leksykograficznego zgodnie z kodem ASCII. Znaczenie parametrów jest następujące:

R — sortowanie w kolejności odwrotnej

+n — liczba naturalna określająca kolumnę, od której tekst jest brany pod uwagę przy sortowaniu (wartość domyślna wynosi 1, co oznacza, że cały wiersz jest traktowany jako klucz).

POLECENIA WSADOWE

DOS pozwala na automatyczne wykonywanie sekwencji poleceń zapamiętywanych w tzw. pliku wsadowym. Plik wsadowy stanowi odpowiednik makropoleceń stosowanych w innych systemach operacyjnych. Polecenie wsadowe ma format:

[d:] filename [parameters]

gdzie plik o nazwie **filename** ma rozszerzenie nazwy **BAT**. W pliku **filename.BAT** mogą znajdować się wszystkie polecenia wewnętrzne i zewnętrzne, używane w zwykłym trybie pracy oraz dodatkowo polecenia **ECHO, FOR, GOTO, IF, SHIFT, PAUSE** i **REM**.

Polecenia wsadowe tworzy się za pomocą dowolnej metody tworzenia plików tekstowych, np. standardowym edytorem **EDLIN**. Polecenia zawarte w pliku wsadowym wykonuje się sekwencyjnie jedno po drugim, o ile nie została wykonana jedna z instrukcji **FOR, GOTO, IF** lub nie istnieje jakaś szczególna sytuacja, np. interwencja użytkownika przez naciśnięcie klawisza **CTRL-BREAK**. Mechanizm poleceń wsadowych umożliwia stosowanie parametrów. W tekście pliku wsadowego parametry formalne oznacza się przez **%0, %1, ..., %9**. Przy realizacji polecenia wsadowego parametry formalne (począwszy od **%1**) są zastępowane tekstowo przez kolejne parametry aktualne (ciągi znaków). Parametr **%0** zawsze jest zastępowany przez **[d:]filename**, tj. nazwę pliku wsadowego.

W trakcie wykonywania poleceń zawartych w pliku wsadowym, aktualnie wykonywane polecenie i zarazem całą sekwencję można przerwać klawiszem **CTRL-BREAK**. Jednak nie wszystkie programy reagują na **CTRL-BREAK**. Plik wsadowy nie może zawierać jako poleceń odwołań do plików wsadowych, z wyjątkiem sytuacji, gdy jest to ostatnie polecenie w pliku.

W systemie PC-DOS specjalną rolę pełni plik wsadowy **AUTOEXEC.BAT**. Przy każdym ładowaniu systemu procesor poleceń sprawdza, czy **AUTOEXEC.BAT** znajduje się w skorowidzu systemowym na dysku, z którego system jest ładowany. Jeśli tak, to zostaje wykonana sekwencja poleceń zawartych w tym pliku.

Poniżej przedstawiono krótki opis poleceń, które mogą występować tylko w plikach wsadowych. Wszystkie te polecenia **(ECHO, FOR, GOTO, IF, SHIFT, PAUSE** i **REM)** znajdują się w części rezydentnej systemu DOS.

ECHO [ON | OFF | message]

Wartością domyślną jest **ON**; wtedy kolejne polecenia z pliku wsadowego są wyświetlane przed wykonaniem (łącznie z wartościami aktualnymi parametrów). Wykonanie polecenia **ECHO OFF** powoduje nie wyświetlanie wykonanych poleceń, a polecenie **ECHO message** wyświetla komunikat **message** niezależnie od tego, czy obowiązuje aktualnie parametr **ON** czy **OFF**.

FOR %%variable IN (set) **DO** command

gdzie: **%%variable** jest nazwą zmiennej, **set** — listą wartości, **command** — poleceniem systemu PC-DOS. Polecenie to umożliwia iteracyjne wykonanie polecenia systemowego w pliku wsadowym. W każdym cyklu iteracji **%%variable** przyjmuje kolejną wartość z listy wartości **(set)** i wykonuje się polecenie **command**.

Polecenia FOR nie mogą być zagnieżdżane. Przykładowo, polecenie:

FOR %%A IN (A: B:) DO DIR %%A

jest równoważne wykonaniu dwóch poleceń:

DIR A:
DIR B:

GOTO label
...
:label

Polecenie **GOTO** umożliwia odejście od ściśle sekwencyjnego wykonywania poleceń i przekazanie sterowania do polecenia, które w poprzedzającej je linii ma etykietę: **:label**.

IF [NOT] condition command

Polecenie **IF** umożliwia warunkowe wykonywanie poleceń. Jeśli po **IF** nie występuje warunek **NOT**, to polecenie **command** jest wykonywane wtedy i tylko wtedy, gdy warunek **condition** przyjmuje wartość **true** (jeśli występuje **NOT**, to warunek musi przyjmować wartość **false**). Warunek można określić w następujący sposób:

1. ERRORLEVEL number

Warunek przyjmuje wartość **true**, jeśli polecenie wykonywane przed **IF** ma kod zakończenia nie mniejszy od **number**.

2. string1 == string2

Warunek przyjmuje wartość **true**, jeśli oba ciągi znaków **string1** i **string2** są identyczne.

3. EXIST filespec

Warunek przyjmuje wartość **true**, jeśli wyspecyfikowany plik znajduje się na podanej stacji dyskowej. Nie dopuszcza się podawania ścieżki dostępu w specyfikacji pliku.

PAUSE [remark]

Polecenie to powoduje wyświetlenie komunikatu **remark**, wyświetlenie komunikatu **"Strike key when ready..."**, a następnie zawieszenie wykonywania polecenia wsadowego.

Spis treści rocznika

INFORMATYKA 1986

ARTYKUŁY PROBLEMOWE

	nr	str.
ABRAMOWICZ WITOLD — Modularny system wyszukiwawczy	1	6
AMBROZIAK JACEK — Systemy ekspertowe		
Część 1	11—12	19
BUDZIANOWSKI LESŁAW, WIETRZYCH JERZY, KWIATKOWSKI JAN — Komputer ODRA 1305 w Międzyuczelnianej Sieci Komputerowej	7—8	16
BIELECKI JAN — Błędy implementacji w kompilatorze języka C firmy Supersoft	1	10
BIELECKI JAN — Zaawansowane konstrukcje języka C	2—3	14
BJØRNER DINES — Rola i zakres definicji Ady		
Część 1	5	1
Część 2	6	17
CELLARY WOJCIECH — Systemy wielomikroprocesorowe		
Część 1	11—12	1
COFTA PIOTR — Norma IEEE — języki asemblerowe dla mikroprocesorów	5	11
DEMINET JAROSŁAW — Cm* — przykład komputera wieloprocesorowewgo o strukturze hierarchicznej	9—10	4
DWORAKOWSKI WALDEMAR — Komputery osobiste IBM PC		
Część 1	7—8	30
Część 2	9—10	15
Część 3	11—12	11
DWORAKOWSKI WALDEMAR — Techniki testowania układów a analiza sygnatur	7—8	40
FABER ROMAN, OSTROWSKI MICHAŁ — UPT — uniwersalny procesor do redagowania tekstów dla MERY 400	4	5
GECOW ANDRZEJ — Litery polskie dla IBM PC	9—10	9
GRABOWICZ ROMAN, ZALEWSKI JANUSZ — System operacyjny PC-DOS		
Część 1	7—8	32
Część 2	9—10	12
Część 3	11—12	13
GRZEGORZEWICZ PAWEŁ — Mechanizm półautomatycznego generatora poleceń dla systemu operacyjnego RSM-11M	5	17
ISZKOWSKI WACŁAW — Architektura systemów n-mikroprocesorowych	7—8	12
JANCZUR ANDRZEJ T. — Rozwiązywanie pasmowych układów równań liniowych na mikrokomputerach	4	9
KARCZMARCZUK JERZY — Język programowania Icon		
Część 1	2—3	11
Część 2	4	13
Część 3	5	14
KLEIBER MICHAŁ, LEŚNY MACIEJ, SZUNIEWICZ ROMUALD — Komputery osobiste w zastosowaniach profesjonalnych		
Część 1	2—3	27
Część 2	4	11
KLEIBER MICHAŁ, SARAN MICHAŁ — Doświadczenia z użytkowania Fortranu na mikrokomputerze IBM PC	7—8	34
KOTULSKI LESZEK — Monitor jako narzędzie strukturalizacji programów współbieżnych		
Część 1	2—3	27
Część 2	4	16
KRÓLIKOWSKI ZBYSZKO — Generowanie planów wykonywania transakcji w systemach rozproszonych baz danych		
Część 1. Podstawowe pojęcia i założenia	6	4
Część 2. Przegląd metod	7—8	20
Część 3. Nowy kierunek badań	9—10	20
MACIOŁ ANDRZEJ, STAWOWY ADAM — Środki dla przetwarzania tekstów — historia i dzień dzisiejszy	5	20
MAĆKÓW PAWEŁ — Komunikacja szeregowa w mikrokomputerach IBM PC i kompatybilnych	11—12	15
MARKOWSKI MACIEJ A. — Logiczna organizacja dysku w systemie operacyjnym PC-DOS	7—8	38
MICHALIK KRZYSZTOF — Programowanie logiczne	5	7
NOSOWSKI WIESŁAW, SZOMAŃSKI BOLESŁAW — Pakiet symulacyjny DESA	1	14
OWCZARCZYK JACEK, STOLARSKI MACIEJ, WOŹNIAK EWA — Architektura współbieżnych systemów przetwarzania obrazów	7—8	8

	nr	str.
PIERZCHAŁA EDMUND — Interakcyjne środowisko języka Lisp		
Część 1	7—8	5
Część 2	9—10	18
PERYCZ KRZYSZTOF — Mikrokomputerowy system operacyjny S1		
Część 1	5	5
Część 2	6	14
PŁOSKI ZDZISŁAW — Współpraca LOGO Sinclaira ze stacją dyskietek	6	9
POSTÓŁ MARIUSZ — Modyfikacja procesu kompilacji PASCALA/MT+	2—3	32
PYLE IAN C. — Pakiet do specyfikacji programów w Adzie	1	1
SIKORA MAREK — Modułowy system mikroprocesorowy kompatybilny z systemem Mikroster	1	12
STOKŁOSA JANUSZ — Abraham Stern — pierwszy polski konstruktor maszyn liczących	2—3	31
STOKŁOSA JANUSZ — Z dziejów sterowania	7—8	24
SUKIENNIK JERZY — Rozrachunek gospodarczy zakładowego ośrodka informatyki	1	18
SZKARADNIK ZBIGNIEW — Operacje zmiennoprzecinkowe w języku Forth	2—3	17
SZKARADNIK ZBIGNIEW — Rekurencja w języku Forth	4	10
SZKARADNIK ZBIGNIEW — Grafika żółwia w języku Forth	6	11
TUZIEMSKI RYSZARD — Język Ratfor dla mikrokomputerów MERA 400	1	16
UDRYCKI WŁADYSŁAW — MML — język komunikacji z systemem komutacyjnym	6	1
WICHMAN BRIAN A. — Konwersja oprogramowania na język Ada	4	1
WIRTH NIKLAUS — Zasady wieloprogramowości i ich implementacja w Moduli-2	2—3	1
WRYCZA STANISŁAW — Przesłanki i uwarunkowania modelowania konceptualnego	11—12	22
ZAKHAROV VASILII — Komputery osobiste oraz indywidualne stanowiska robocze	7—8	26
ZAKHAROV VASILII — Lokalna sieć komputerowa ETHERNET		
Część 1	9—10	1
Część 2	11—12	4
ZABRODZKI JAN — Stan obecny i perspektywy rozwoju układów LSI i VLSI	7—8	1
ZAKRĘCKI JERZY — Abstrakcje w programowaniu		
Część 1	1	4
Część 2	2—3	22
ZASKÓRSKI PIOTR, KASPRZAK EWA — Technologia wytwarzania oprogramowania użytkowego z komputerowym wspomaganiem	6	20
ZIELIŃSKI KRZYSZTOF, INDULSKA JADWIGA, KRASOWSKI ROMAN — Lokalna sieć komputerowa UMMLAN-2	11—12	8
Zmiany i uzupełnienia w Moduli-2 — **oprac. JZ.**	2—3	8
Modula-2 lista implementacji i program wzorcowy — **oprac. JZ.**	2—3	9
Dwie uogólnione zmiennoprzecinkowe reprezentacje liczb — **Oprac. Mariusz Kuc**	2—3	35
Sterownik sieci Ethernet w systemie Multibus — **oprac. Roman Grabowicz**	4	19
Konstruowanie lokalnych sieci komputerowych według specyfikacji Ethernet — **Roman Grabowicz**	7—8	43

SAMOTESTY

IV/C. Organizacja i przetwarzanie plików — **Adam B. Empacher, Jerzy L. Rossowski**	1	19

Z KRAJU

10 lat Zakładu Obliczeniowego ETOB w Lublinie — **R.B.**	1	20
Techniki komputerowe w zarządzaniu produkcją	1	21
Zastosowanie mikrokomputerów szansą usprawnienia zarządzania gospodarką — **Stefan Rakowski**	2—3	35
Dolnośląski Oddział Towarzystwa Informatycznego — **WI.**	2—3	37
Pracownie komputerowe SEP i „Horyzontów Techniki".	2—3	38
Działalność Komitetu ds. Systemu CAMAC — **J. Zal.**	2—3	38
Konkurs Polskiego Towarzystwa Informatycznego na najlepsze prace magisterskie z informatyki — rozpoczęty — **Barbara Osuchowska**	6	23
Szybki system operacyjny DOS	6	23
INSTYTUT INFORMATYKI Politechniki Warszawskiej	9—10	24
MIKROKOMPUTERY '85	11—12	24

ZE ŚWIATA

Mikromyszy na start! — **Ryszard K. Kott**	1	22
Zmierzch pomysiu Gutenberga — **Wacław Iszkowski**	1	25
Quattro — nowy komputer osobisty ICL — **Władysław Klepacz**	1	26

	nr	str.
Przetwarzanie współbieżne w Moduli-2 — **Maciej Adamczyk**	2—3	40
Ochrona programów przed kopiowaniem w komputerach osobistych — **Stanisław Chmielewski**	2—3	41
Metastabilność systemów VME i MULTIBUS II — **Anna Czarnecka**	2—3	43
Nowa generacja 32-bitowych systemów wieloprocesorowych z wykorzystaniem magistrali MULTIBUS — **Anna Czarnecka**	2—3	44
Zmierzch mikroprocesorów 8-bitowych	2—3	44
Historia i rozwój mikroprocesorów — **oprac. Mariusz Kuc**	4	22
Sir Clive Sinclair o trzeciej rewolucji przemysłowej — **oprac. Roma Barlak**	4	25
Standaryzacja języka BASIC — **Wacław Iszkowski**	4	27
Mikrokomputery w Wielkiej Brytanii — **Mariusz Kuc**	5	26
Strukturalizacja Basica — **Mariusz Kuc**	6	26
Transputery — przyszłość techniki komputerowej	6	26
Sytuacja na rynku elementów półprzewodnikowych	6	27
Megabitowe kostki pamięciowe — **Jan Ryżko**	6	28
Podprogramy obsługi przerwań dla IBM PC — **Roman Grabowicz**	7—8	43
European Workshop on Industrial Computer Systems (EWICS) Technical Committee 7 (TC 7)	7—8	50
DATAPOINT i co dalej?	9—10	27

Kto jest kim w IFIP

Pierre Bobillier	4	29
George Glaser	6	26
Ashley Goldsworthy	7—8	51
Kaoru Ando	9—10	29
Robert Piloty	11—12	26

RECENZJE

	nr	str.
BASIC mikroinformatyczny — **Adam B. Empacher**	1	27
Systemy informatyczne zarządzania — **Andrzej Sokołowski**	1	28
Ethernet — książka źródłowa — (yz)	6	29

TERMINOLOGIA

A jednak po polsku! — **Wacław Iszkowski**	1	29
Terminy związane z pamięciami dyskowymi	2—3	45
Trzydziestolecie terminu bajt	4	30
Cardware	4	30
Wylicznik	4	30
Zapożyczenia w polskim słownictwie informatycznym — **Janusz Stokłosa**	5	27
Terminologia definicji formalnej Ady	6	30
Nareszcie ИНФОРМАТИКА = INFORMATYKA	7—8	51
Polskie nazewnictwo znaków kodu ASCII/ISO Część 1	9—10	29
Terminologia lokalnych sieci komputerowych Część 1	11—12	III okł.

POGLĄDY

Z pamiętnika programującego w Basicu — **Zdzisław Płoski**	4	III okł.
Zakleszczenie — **Wacław Iszkowski**	5	23
Informatyka doświadczalna — **Zdzisław Płoski**	7—8	52

KONFERENCJE

II Międzynarodowa Szkoła MIKROKOMPUTER '86	2—3	10
Sieci komputerowe	7—8	15
EUROCAL '87 — Europejska Konferencja Algebry Komputerowej	11—12	26
„Zastosowanie mikrokomputerów w ośrodkach inte"	11—12	26

EGZEMPLARZE ARCHIWALNE czasopisma można nabywać za gotówkę w Klubie Prasy Technicznej w Warszawie, ul. Mazowiecka 12, tel. 27-43-65 lub zamówić w Dziale Handlowym Wydawnictwa NOT SIGMA, ul. Bartycka 20, skr. poczt. 1004, 00-950 Warszawa, tel. 40-37-31, na rachunek dla instytucji lub za zaliczeniem pocztowym dla osób fizycznych.

UNIVERSITAS

STUDENCKA SPÓŁDZIELNIA PRACY

Oferuje usługi z zakresu projektowania
i programowania systemów informatycznych
oraz obsługi eksploatacyjnej
i technicznej komputerów

Zlecone prace wykonują studenci
wyższych lat studiów
pod nadzorem i przy udziale pracowników
warszawskich wyższych uczelni.

Zgłoszenia w godz. 7.30—16.00 przyjmuje
Zakład Usług Informatycznych SPP UNIVERSITAS
Warszawa, ul. Nowowiejska 37, p. 2, tel. 25-56-28

EO/132/K/86

CONPOL

oferuje minikomputery
CONPOL PC/XT/AT
kompatybilne z IBM
z pełnym wyposażeniem
peryferyjnym.
Zestawy na życzenie klienta.
Serwis gwarancyjny i pogwarancyjny
z magazynem konsygnacyjnym.
Dostawy natychmiastowe.

Bliższe informacje:
P.Z. CONPOL
71-581 Szczecin
ul. Pszczelna 7
tel. 23-39-00, 23-39-65
lub
66-520 Dobigniew
ul. Dzierżyńskiego 1, tel. 123
lub
Centrum Minikomputerowe SYSTEM
Warszawa, ul. Wolska r. Młynarskiej
(przejście podziemne) tel. 32-80-93

EO/1302/K/86

ODiTK

Gdańsk

mikrokomputery

doradztwo

ODiTK

może poprawić

trafność decyzji

EO/524/K/86

Wznowienie działania następuje przez naciśnięcie dowolnego klawisza z wyjątkiem **CTRL-BREAK** — w tym wypadku przetwarzanie polecenia wsadowego zostaje przerwane.

REM remark

Polecenie to powoduje wyświetlenie tekstu **remark**.

INNE POLECENIA

CLS

Polecenie wewnętrzne **CLS** powoduje oczyszczenie ekranu na standardowym urządzeniu wyjściowym.

EXE2BIN file [d:][path][filename[.ext]]

Polecenie to przekształca plik wygenerowany przez konsolidator (z rozszerzeniem nazwy **EXE**) na plik o formacie odpowiadającym poleceniom z rozszerzeniem nazwy **COM**. Plik przekształcany podaje się jako pierwszy parametr polecenia.

PAWEŁ MAĆKÓW
Instytut Cybernetyki Technicznej
Politechnika Wrocławska

Ogłoszenia • Ogłoszenia • Ogłoszenia • Ogłoszenia

BIURO USŁUG KOMPUTEROWYCH. Pośrednictwo sprzedaży mikrokomputerów, części zamiennych. Warszawa, tel. 41-44-48.

EO/272/K/86

Ogłoszenia • Ogłoszenia • Ogłoszenia • Ogłoszenia

EGZEMPLARZE ARCHIWALNE CZASOPISMA można nabywać za gotówkę w Klubie Prasy Technicznej w Warszawie, ul. Mazowiecka 12, tel. 27-43-65 oraz w Dziale Handlowym Wydawnictwa, ul. Bartycka 20, skr. poczt. 1004, 00-950 Warszawa, tel. 40-37-31, na rachunek dla instytucji lub za zaliczeniem pocztowym dla osób fizycznych.

Komunikacja szeregowa w mikrokomputerach IBM PC i kompatybilnych

Duża popularność i stosunkowo łatwa dostępność mikrokomputerów kompatybilnych z IBM PC/XT wpływa na coraz szerszy zakres ich zastosowań — od narzędzia do obliczeń inżynierskich, przez urządzenie do redagowania tekstów i bazę danych, do inteligentnej końcówki w sieciach mikrokomputerowych.

O ile zastosowanie mikrokomputera w dziedzinach wymagających tylko dobrze udokumentowanego oprogramowania nie nastręcza specjalnych trudności, o tyle kwestia wykorzystania istniejącego sprzętu i jego współpracy z IBM PC jest znacznie poważniejsza, zwłaszcza że firmy oferujące mikrokomputer często nie dostarczają dokumentacji sprzętowej.

Problemem najczęściej spotykanym jest wykorzystanie złącza RS232C w mikrokomputerze do komunikacji według określonych założeń, np. dołączenie do komputera drukarki D-180 lub modemu. Rozwiązanie zagadnień tego rodzaju jest możliwe tylko po przestudiowaniu pełnej dokumentacji sprzętowej i programowej.

Poniższy artykuł dostarcza informacji o wykorzystaniu łącza szeregowego w IBM PC/XT, AT i komputerach kompatybilnych.

STRUKTURA SPRZĘTOWA ŁĄCZA SZEREGOWEGO

Interface szeregowy znajduje się na jednym z pakietów dodatkowych, umieszczonych w gniazdkach płyty głównej. Złącze szufladowe 25-stykowe jest wyprowadzone na maskownicę pakietu, przy czym nie ma reguły co do stosowanego typu złącza — może ono być terminalowe (wtyk) albo modemowe (gniazdo). Rolę konwertera danych z magistrali na postać szeregową i odwrotnie pełni układ UART (ang. Universal Asynchronous Receiver-Transmitter) typu 8250, produkowany m.in. przez firmę National Semiconductor, mało popularny w sprzęcie stosowanym w Polsce. Układ 8250, poza nadajnikiem i odbiornikiem asynchronicznej transmisji szeregowej, zawiera również programowany generator szybkości, transmisji i prosty priorytetowy sterownik przerwań. Dopasowanie poziomów sygnałów do standardu zalecenia V.24 CCITT realizują układy 1488/1489.

Konfiguracja układu 8250 w mikrokomputerze IBM PC jest następująca:

- adresy w przestrzeni we-wy procesora 8088
03F8H...03FEH — port COM1
02F8H...02FEH — port COM2
(program BIOS może obsłużyć maksymalnie cztery porty RS 232C, podano adresy dwóch pierwszych),
- generowanie przerwań na liniach INT2...INT7 systemowego sterownika przerwań 8259A (standardowo system operacyjny nie używa przerwań z kanału szeregowego, jednakże część oprogramowania np. Basic i Crosstalk wykorzystuje przerwanie INT4 od kanału 1 i INT3 od kanału 2; linia żądania przerwania z układu UART może być dołączona do jednej z linii INT2...INT7 za pomocą zworki na pakiecie),

Tabela 1. Wybrane linie styku szeregowego

Nr linii styku S2	Nr końcówki na złączu	Nazwa sygnału	Opis sygnału
103	2	TxD — wy	Dane nadawane z IBM PC
104	3	RxD — we	Dane odbierane
105	4	RTS — wy	Żądanie nadawania do modemu
106	5	CTS — we	Gotowość modemu do przyjęcia danych
107	6	DSR — we	Sygnalizacja włączenia modemu
108	20	DTR — wy	Sygnalizacja włączenia komputera
109	8	RLSD — we	Wskaźnik poziomu odbieranego sygnału z linii telefonicznej, stan niski oznacza zgubienie nośnej
125	22	RI — we	Sygnał żądania nawiązania łączności (dzwonek) z linii telefonicznej do modemu
102	7	GND	Ziemia sygnałowa

● obsługa linii styku S2 według tabeli 1.

Poziomy napięciowe podanych sygnałów obsługiwanych przez mikrokomputer są następujące:

● na wejściu
"1" — −3...−18 V
"0" — +3...+18 V

● na wyjściu
"1" — −12 V
"0" — +12 V

Zastosowany w IBM PC układ UART umożliwia obsługę praktycznie każdego urządzenia o transmisji szeregowej asynchronicznej, zarówno DCE (ang. Data Communications Equipment, np. modem), jak i DTE (ang. Data Transmission Equipment, np. dalekopis). Typ urządzenia można wybrać, konstruując odpowiedni kabel. Przykładowe połączenia zilustrowano w tabeli 2. Ze względu na dużą różnorodność złącz szeregowych w wyrobach różnych firm konieczne jest dokładne zapoznanie się z ich dokumentacją przed wykonaniem połączenia.

Tabela 2. Sposób połączenia mikrokomputera IBM PC z niektórymi urządzeniami o asynchronicznej transmisji szeregowej RS 232C

IBM PC		Modem	IBM PC		Drukarka
2 — TxD	→	TxD — 2	2 — TxD	→	TxD — 2
3 — RxD	←	RxD — 3	5 — CTS	←	CTS — 5
4 — RTS	→	RTS — 4	6 — DSR	→	DSR — 6
5 — CTS	→	CTS — 5	7 — GND	—	GND — 7
6 — DSR	—	DSR — 6			
7 — GND	→	GND — 7			
8 — CD	←	CD — 8			
20 — DTR	→	DTR — 20			
22 — RI	←	RI — 22			

IBM PC		Drukarka o protokole ETX/ACK (wymaga specjalnego oprogramowania)	IBM PC		Inny komputer
2 — TxD	→	TxD — 2	2 — TxD	→	RxD — 3
3 — RxD	←	RxD — 3	3 — RxD	←	TxD — 2
4 — RTS ┐			4 — RTS ┐		RTS — 4
5 — CTS ←			5 — CTS ←		CTS — 5
6 — DSR ←			6 — DSR ←		DSR — 6
20 — DTR ┘			20 — DTR ┘		DTR — 20
7 — GND	—	GND — 7	7 — GND	—	GND — 7

OPROGRAMOWANIE KANAŁU SZEREGOWEGO

Użytkownik mikrokomputera IBM PC z zainstalowanym systemem operacyjnym MS-DOS ma do dyspozycji trzy możliwości programowej obsługi kanału szeregowego:

— bezpośrednie odwołanie do układu UART,
— wywołania programu ROM BIOS (INT 14H),
— wywołania procedur systemu operacyjnego (INT 21H), od wersji 2.0.

Bezpośrednia obsługa portów we-wy układu 8250

Metoda ta umożliwia wykorzystanie wszystkich możliwości układu UART użytego w mikrokomputerze, ogranicza jednak zakres zastosowań programu do urządzeń w pełni kompatybilnych z IBM PC/XT.

Układ 8250 zawiera 9 rejestrów dostępnych dla użytkownika i zajmuje 7 adresów w przestrzeni we-wy mikroprocesora 8088 (tab. 3—8). Funkcje rejestrów o adresach względnych 00 i 01 są podwójne (tab. 3) i zależą od stanu bitu DLAB (bit 7 rejestru 03 — LCR). Adresy rejestrów podano w odniesieniu do adresu bazowego (normalnie 03F8H dla portu COM1, 02F8H dla COM2).

Zmienne systemowe o adresach 00400H (COM1), 00402H (COM2), 00404H (COM3) i 00406H (COM4) zawierają adresy bazowe układów 8250 na pakietach komunikacyjnych. Gdy zmienna zawiera liczbę 0000H — nie istnieje odpowiadający jej pakiet komunikacyjny. System operacyjny MS-DOS obsługuje tylko dwa porty komunikacyjne w standardowej konfiguracji: COM1: (AUX:) oraz COM2:.

Tabela 3. Funkcje rejestrów 00 (THR/RBR lub DLL) i 01 (IER lub DLM) układu 8250

Bity	Funkcja
0—7	**Gdy bit DLAB = 0**, zawiera dane nadawane lub odbierane (THR/RBR); gdy DLAB = 1, zawiera mniej znaczący bajt dzielnika częstotliwości (DLL)
0—7	**Gdy bit DLAB = 1**, rejestr zawiera bardziej znaczący bajt dzielnika częstotliwości (DLM), **gdy bit DLAB = 0**, znaczenie bitów się zmienia i rejestr zawiera maskę przerwań (IER)
0	**ERBFI** (ang. Enable Receiver Buffer Full Interrupt); bufor odbiornika pełny — priorytet 2
1	**ETBEI** (ang. Enable Transmitter Buffer Empty Interrupt), bufor nadajnika pusty — priorytet 3
2	**ELSI** (ang. Enable Line Status Interrupt), stan odbiornika — priorytet 1 (najwyższy)
3	**EDSSI** (ang. Enable Data Set Status Interrupt), stan linii modemowych — priorytet 4
4—7	Nieużywane, zerowane przy odczycie.

Tabela 4. Funkcje rejestru 02 (IIR) układu szyn (identyfikator przyczyny przerwania)

Bity (tylko do odczytu)	Funkcja
0	Ustawiany, gdy wystąpiło przerwanie; kasowany przy odczycie odpowiedniego rejestru
1—2	Kod przyczyny przerwania (odpowiada przyczynie o najwyższym priorytecie): 00 — stan linii modemowych CTS, DSR, RI, RLSD (odczyt rejestru stanu modemu MSR) 01 — bufor nadajnika pusty (odczyt rejestru IIR lub zapis danych do bufora danych nadawanych) 10 — bufor odbiornika pełny (odczyt danych z bufora odbiornika) 11 — stan odbiornika tj. błąd parzystości, błąd zgubienia znaku, błąd ramki znaku, lub sygnał BREAK (odczyt rejestru stanu linii LSR)
3—7	Kasowane

Tabela 5. Funkcje rejestru 03 (LCR) układu 8250 (sterowanie parametrami transmisji)

Bity	Funkcja
0—1	**WLS** (ang. Word Length Select), długość słowa danych: 00 — 5 bitów 01 — 6 bitów 10 — 7 bitów 11 — 8 bitów
2	**STB** (ang. STop Bits), liczba bitów STOP: 0 — 1 bit 1 — 2 bity
3	**PEN** (ang. Parity ENable), generowanie parzystości: 0 — zabroniona 1 — dozwolona
4	**EPS** (ang. Even Parity Select), rodzaj badania lub stan bitu parzystości: 0 — nieparzystość lub logiczne 0 1 — parzystość lub logiczne 1
5	**Stick Parity**, ustalenie wartości bitu parzystości na bit określony jako rodzaj badania, o ile generowanie parzystości jest dozwolone, tzn. dodanie jako bitu parzystości bitu o wartości ustalonej przez EPS
6	**Set Break**, ustawienie tego bitu powoduje wymuszenie stanu logicznego 0 w linii (BREAK) aż do momentu skasowania bitu 6
7	**DLAB** (ang. Divisor Latch Access Bit), ustawienie tego bitu udostępnia rejestry 0 i 1 jako stałą dzielnika częstotliwości; powrót do normalnej funkcji następuje po skasowaniu bitu 7

Układ 8250 ma możliwość zaprogramowania szybkości transmisji przez wpisanie odpowiedniej wartości do zatrzasków licznika (rejestry 00 i 01, gdy bit DLAB jest ustawiony). Wartości wpisywane do dzielników są zależne od częstotliwości zegara (1,8432 MHz w IBM PC) i odpowiadają szesnastokrotnej szybkości transmisji (tabela 9).

Tabela 6. Funkcje rejestru 04 (MCR) układu 8250 (sterowanie liniami modemu). Ustawienie odpowiedniego bitu w tym rejestrze powoduje wymuszenie stanu nieaktywnego (+12 V) na liniach współpracujących z modemem

Bity	Funkcja
0	Linia DTR
1	Linia RTS
2	Końcówka OUT1
3	Końcówka OUT2
4	Ma znaczenie specjalne i jego ustawienie powoduje przejście układu w stan testowania. Układ nadajnika jest wewnętrznie połączony z rejestrem odbiornika, co umożliwia sprawdzenie poprawnej pracy układu 8250. Przy normalnej pracy bit 4 powinien być skasowany.
5—7	Nieużywane, kasowane przy odczycie.

Tabela 7. Funkcje rejestru 05 (LSR) układu 8250 (stan linii nadajnika i odbiornika)

Bity	Funkcja
0	DR (ang. Data Ready), bufor odbiornika zawiera odebrany znak; bit jest kasowany po odczycie znaku z rejestru RBR (o adresie 0)
1	OE (ang. Overrun Error) wskazuje, że dana w buforze odbiornika nie została odczytana do momentu odebrania następnego znaku z linii; poprzednio odebrany znak jest gubiony
2	PE (ang. Parity Error) — wskaźnik błędnej parzystości odebranego znaku
3	FE (ang. Framing Error), odebrany znak nie miał odpowiedniego bitu STOP
4	BI (ang. Break Interrupt) oznacza, że linia była w stanie logicznego zera przez czas dłuższy od czasu trwania pełnego znaku łącznie z bitami START i STOP (wystąpił sygnał BREAK)
5	THRE (ang. Transmitter Holding Register Empty) sygnalizuje, że bufor nadajnika może przyjąć następny znak; jest kasowany po zapisie danych do nadajnika
6	TSRE (ang. Transmitter Shift Register Empty) wskazuje, że nadajnik zakończył wysłanie informacji i rejestr przesuwny nadajnika jest pusty
7	Nieużywany, kasowany przy odczycie

Tabela 8. Funkcje rejestru 06 (MSR) układu 8250 (stan linii wejściowych współpracujących z modemem)

Bity	Funkcja (ustawienie jednego z bitów 0—3 może wywołać przerwanie od modemu, a bity 4—7 odzwierciedlają stan linii modemu)
0	DCTS (ang. Delta Clear To Send), wejście CTS zmieniło stan od ostatniego odczytu rejestru MSR
1	DDSR (ang. Delta Data Set Ready), wejście DSR zmieniło stan od ostatniego odczytu rejestru MSR
2	TERI (ang. Trailing Edge Ring Indicator), linia sygnału dzwonka z modemu zmieniła stan z logicznego 1 na logiczne 0
3	DRLSD (ang. Delta Received Line Signal Detect), wejście RLSD zmieniło stan od ostatniego odczytu rejestru MSR
4	CTS (ang. Clear To Send)
5	DSR (ang. Data Set Ready)
6	RI (ang. Ring Indicator)
7	RLSD (ang. Received Line Signal Detect)

Tabela 9. Standardowe szybkości transmisji

Szybkość [bd]	Dzielnik częstotliwości podstawowej
50	2304
75	1536
110	1047
134,5	857
150	768
300	384
600	192
1200	96
2400	48
4800	24
9600	12
19200	6
38400	3
56000	2

Wykorzystanie procedur zawartych w programie ROM BIOS

Procedury obsługujące kanał transmisji szeregowej są dostępne za pośrednictwem programowego przerwania INT 14H. Rodzaj wykonywanej funkcji podaje się w rejestrze AH, a pełen zestaw funkcji oraz sposób ich wywoływania przedstawiono poniżej:

```
MOV DX,nr_karty      ;0, 1, 2 lub 3
MOV AH,nr_funkcji    ;0, 1, 2 lub 3
MOV AL,parametr      ; tylko w wywoływaniach 0 i 1
INT 14H
```

Funkcja 0 — inicjalizacja układów na pakiecie transmisji. Wybór trybu pracy następuje według zawartości rejestru AL, przy czym poszczególne bity mają następujące oznaczenia:

```
bit          — 7 6 5 4 3 2 1 0,
oznaczenie   — S2 S1 S0 PE EP S L1 L0,
```

gdzie ciąg S2, S1, S0 oznacza szybkość transmisji według poniższego schematu:
000 — 110 bodów
001 — 150 bodów
010 — 300 bodów
011 — 600 bodów
100 — 1200 bodów
101 — 2400 bodów
110 — 4800 bodów
111 — 9600 bodów
PE=1 oznacza zezwolenie na generowanie bitu parzystości,
EP=1 — bit parzystości,
EP=0 — bit nieparzystości,
S=1 — 2 bity STOP,
S=0 — 1 bit STOP,
L1, L0 — kod długości znaku według następującego schematu:
00 — 5 bitów
01 — 6 bitów
10 — 7 bitów
11 — 8 bitów.
W wyniku rejestr AX zawiera informację o stanie nadajnika, odbiornika i linii modemowych (jak w funkcji 3).

Funkcja 1 — wysłanie znaku o kodzie podanym w rejestrze AL. Jeśli znaku nie można wysłać, to przy powrocie z wywołania ustawiany jest bit 7 rejestru AH. Bity 6 i 5 rejestru AH odpowiadają stanowi rejestrów wyjściowych układu UART po wysłaniu znaku (patrz funkcja 5). Zawartość rejestru AL nie jest zmieniana.

Funkcja 2 — odebranie znaku z linii. Jeśli znak odebrano, to rejestr AL zawiera jego kod. Zawartość rejestru AH odzwierciedla stan odbiornika po wykonaniu operacji (jak w funkcji 3).

Funkcja 3 — odczytanie stanu nadajnika, odbiornika i linii sterujących transmisją. Rejestr AH zawiera informacje o stanie nadajnika-odbiornika układu UART. Znaczenie poszczególnych bitów jest następujące:
7 — time-out; operacja nadawania-odbioru nie została przeprowadzona z powodu niegotowości urządzenia zewnętrznego
6 — wyjściowy rejestr przesuwny nadajnika jest pusty (TxSRE)
5 — bufor nadajnika jest pusty (TxHRE)
4 — odebrano sygnał BREAK z linii (BD)
3 — błąd ramki znaku (Framing Error)
2 — błąd parzystości (Parity Error)
1 — błąd, zgubienie znaku (Overrun Error)
0 — dane w odbiorniku są gotowe do odczytania.

Rejestr AL zawiera informację o stanie linii sterujących współpracą z modemem. Dokładny opis znaczenia poszczególnych bitów jest identyczny jak w rejestrze 06 (MSR) układu UART 8250:

7 — RLSD
6 — RI
5 — DSR
4 — CTS
3 — Delta RLSD
2 — Trailing Edge RI
1 — Delta DSR
0 — Delta CTS

Przy wykorzystaniu procedur programu ROM BIOS użytkownik może sterować parametrami transmisji oraz

odczytywać stan sterujących linii modemowych. Nie istnieje jednak możliwość sterowania liniami RTS i DTR oraz wykorzystania przerwań generowanych przez układ UART. Wydaje się, że wykorzystywanie w programach użytkowych wywołań programu BIOS jest rozwiązaniem najbardziej optymalnym ze względu na fakt, że większość mikrokomputerów, szesnastobitowych z systemem operacyjnym MS-DOS korzysta z programu BIOS o identycznej strukturze jak IBM PC/XT, nawet, jeśli użyte układy nie są odpowiednikami 8250. Większość firmowego oprogramowania wykorzystuje właśnie ten sposób komunikacji.

Wywołania systemu operacyjnego MS-DOS

Jest to najprostsza metoda obsługi kanału RS 232C, pozwalająca na dwukierunkową transmisję danych z jedną ze standardowych szybkości i w jednym ze standardowych formatów. Użytkownik jest pozbawiony możliwości kontroli stanu linii sterujących transmisją i nie może wykorzystywać przerwań do obsługi nadajnika-odbiornika układu 8250.

Wyboru parametrów transmisji dokonuje się przy użyciu programu systemowego MODE, wywołując go w następujący sposób:

MODE COMn: v, p, d, s [,P]

gdzie poszczególne parametry mają następujące znaczenie:
n={1 — 2} — nr kanału szeregowego
v={110 — 150 — 300 — 600 — 1200 — 2400 — 4800 — 9600} — szybkość transmisji w bitach/s
d={7 — 8} — długość słowa danych w bitach
p={O — E — N} — badanie parzystości, O — nieparzystość, E — parzystość, N — bez badania parzystości
s={1 — 2} — liczba bitów STOP
P — podanie tego parametru jako ostatniego powoduje, że system operacyjny czeka dowolnie długo na gotowość odbiornika; jego pominięcie powoduje wygenerowanie komunikatu o błędzie po kilkudziesięciosekundowej niegotowości odbiornika sygnalizowanej nieaktywnym stanem linii CTS i DSR.

Po odpowiednim ustaleniu parametrów transmisji można użyć nazw COM1 (lub AUX) oraz COM2 zamiast nazw plików, w komendach systemu MS-DOS, np.:
copy test.asm com1 — wysłanie pliku tekstowego przez kanał RS 232C
copy test.com aux/b — wysłanie pliku binarnego,
dir > com2 — wysłanie listy plików w skorowidzu na drukarkę szeregową (kanał nr 2).

Programiści używający języka asemblera mają do dyspozycji dwie podstawowe funkcje obsługujące kanał szeregowy COM1: „Czytaj znak" i „Pisz znak". Odczytanie znaku z portu szeregowego realizuje funkcja 3 systemu MS-DOS, wywoływana w następujący sposób:

```
MOV AH,3
INT 21H
```

Po wywołaniu, rejestr AL zawiera kod odebranego znaku. Należy zwrócić uwagę na fakt, że program oczekuje na znak i nie ma możliwości szybkiego sprawdzenia, czy w buforze odbiornika znajdują się ważne dane. Wysłanie znaku realizuje funkcja 4 wywoływana następująco:

```
MOV AH,4
MOV DL,kod_znaku
INT 21H
```

Znak o kodzie podanym w rejestrze DL jest transmitowany przez kanał RS232C, o ile linie CTS i DSR sygnalizują gotowość urządzenia współpracującego. Przerwanie systemowe INT 21H zachowuje stan rejestrów procesora.

Brak gotowości urządzenia współpracującego z IBM PC, sygnalizowany nieaktywnym stanem linii CTS i DSR, nieodebranie znaku w określonym czasie lub błąd w odbieranym znaku powodują komunikat systemowy, np.:

Not ready error writing device COM1:
Abort, Retry, Ignore?

Użytkownik może w takiej sytuacji zakończyć pracę programu (A), ponowić próbę nadawania lub odbioru (R) albo pominąć nadawanie lub odbiór znaku (I).

Należy zwrócić uwagę na fakt, że użytkownik korzystający jedynie z wywołań systemu MS-DOS jest pozbawiony możliwości programowej zmiany parametrów transmisji w czasie pracy programu, jak również kontroli stanu linii sterujących transmisją. Zaletą tego rozwiązania jest natomiast możliwość łatwego przenoszenia oprogramowania na inne mikrokomputery pracujące pod kontrolą systemu MS-DOS (wersje od 1.1. do 3.1).

* * *

W podsumowaniu należy powiedzieć, że użytkownik mikrokomputerów IBM PC/XT, AT i kompatybilnych ma dość szerokie możliwości skonfigurowania pakietu komunikacyjnego i możliwości programowej obsługi kanału transmisji szeregowej. Istnieją trzy poziomy współpracy programu użytkowego z urządzeniami peryferyjnymi wykorzystującymi szeregową transmisję danych:
— obsługa portów układu UART typu 8250,
— wykorzystanie programu ROM BIOS,
— wykorzystanie procedur systemu operacyjnego MS-DOS.

WARUNKI PRENUMERATY NA 1987 R.

Prenumeratorzy zbiorowi — jednostki gospodarki uspołecznionej, instytucje i organizacje społeczne zamawiają prenumeratę dokonując wpłat na blankiecie „polecenie przelewu".

Prenumeratorzy indywidualni — osoby fizyczne zamawiają prenumeratę dokonując wpłaty w UPT lub NBP na blankiecie Wydawnictwa lub blankiecie NBP. Na odwrocie wszystkich odcinków blankietu należy wpisać tytuł czasopisma, okres prenumeraty, liczbę zamawianych egzemplarzy oraz wartość wpłaty.

Wpłacać należy na konto NBP III O/M Warszawa 1036-7490-139-11.

Prenumerata ulgowa — przysługuje wyłącznie osobom fizycznym — członkom SNT, studentom i uczniom szkół zawodowych. Warunkiem prenumeraty ulgowej jest poświadczenie blankietu wpłaty (przed jej dokonaniem) na wszystkich odcinkach pieczęcią Koła SNT, wyższej uczelni lub szkoły.

Sposób zamawiania prenumeraty taki sam jak dla prenumeraty indywidualnej.

Prenumerata ze zleceniem wysyłki za granicę — zamawia się tak jak prenumeratę indywidualną. Dodatkowo należy podać na blankiecie wpłaty nazwisko i dokładny adres odbiorcy. Cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa.

Przedpłaty na prenumeratę przyjmowane są w terminach:

— do 10 listopada na I kwartał, I półrocze i cały rok następny,
— do 28 lutego na II, III, IV kwartał i II półrocze,
— do 31 maja na III, IV kwartał i II półrocze,
— do 31 sierpnia na IV kwartał.

Uwaga!

Wpłaty na dwumiesięczniki przyjmowane są na okresy półroczne lub roczne.

Informacji o prenumeracie udziela — Zakład Kolportażu Wydawnictwa NOT-SIGMA, ul. Bartycka 20, 00-716 Warszawa, lub skr. poczt. 1004, 00-950 Warszawa, tel. 40-00-21 w. 249, 293, 297, 299 oraz 40-35-89 i 40-30-86.

Egzemplarze archiwalne czasopism — można nabyć za gotówkę w Klubie Prasy Technicznej w Warszawie ul. Mazowiecka 12, tel. 27-43-65 lub zamówić w Dziale Handlowym Wydawnictwa, ul. Bartycka 20, skr. poczt. 1004, 00-950 Warszawa, tel. 40-37-31, na rachunek dla instytucji lub za zaliczeniem pocztowym dla osób fizycznych.

Cena miesięcznika INFORMATYKA została ustalona na 150 zł za numer (50 zł — cena ulgowa).

Cena prenumeraty wg cennika					
kwartalna		półroczna		roczna	
normalna	ulgowa	normalna	ulgowa	normalna	ulgowa
450	150	900	300	1800	600

JACEK AMBROZIAK
Środowiskowe Centrum Obliczeniowe
CYFRONET
Instytut Energii Atomowej
Otwock-Świerk

Systemy ekspertowe (I)

Celem niniejszego artykułu jest ogólne przedstawienie tematyki systemów ekspertowych. W pierwszej części znajdują się informacje o powstaniu tej interesującej dziedziny, a także o budowie typowych systemów. W drugiej części przedstawiono nadzieje, jakie wiążą się dziś z tą dziedziną i podstawowe trudności, z jakimi muszą się borykać jej twórcy.

GENEZA

Systemy ekspertowe określa się jako jedną z dziedzin sztucznej inteligencji. Sztuczna inteligencja (SI) — dziś licząca już 30 lat — stawia sobie za cel uchwycenie środkami informatyki skomplikowanych procesów przetwarzania informacji symbolicznej, takich jakie mogłyby stać się podstawą funkcjonowania programów zaprojektowanych do zadań wykonywanych dotąd wyłącznie przez ludzi. Chodzi tu o zadania zazwyczaj stereotypowe — łatwe lub trudne — lecz zawsze związane ze złożonymi postaciami danych i procesów ich obróbki. Przetwarzane symbole są najczęściej nośnikami informacji semantycznej, tzn. wykonuje się na nich działania stosowne do ich znaczenia. W zastosowaniach numerycznych znaczeniem symbolu-liczby jest wyłącznie wartość liczbowa. W dziedzinie sztucznej inteligencji jest inaczej — znaczenie symbolu konstytuują procesy, które się doń stosuje. Bogactwo znaczeń, które można przypisać informacji symbolicznej zwiększa się wraz ze złożonością skojarzonych z nią struktur danych i procedur. Analiza zaś, nawet pozornie bardzo prostych czynności umysłowych człowieka — jeśli jest dostatecznie szczegółowa — dowodzi, jak poważny użytek czyni się ze znaczeń stosowanych pojęć. Twórcy programów SI nieustannie podejmują więc trud doskonalenia metod reprezentacji informacji semantycznej potrzebnej w różnego rodzaju sytuacjach zadaniowych. Chodzi im przy tym o metody, a więc uniwersalne sposoby nacechowane systematycznością, nie zaś o powierzchowne techniki ad hoc.

W pierwszej dekadzie badań interesowano się więc najogólniejszą strukturą rozwiązywania zadań, metodami rozbioru problemu na podproblemy, analizą celów i środków (ang. means-ends analysis), czy heurvstycznie ograniczanym przeglądaniem generowanego kombinatorycznie drzewa stanów — etapów rozwiązywania zadania.

Rezultaty tych badań, aczkolwiek ważne z poznawczego punktu widzenia, zdawały się omijać wszelkie praktyczne zastosowania. Dziedzina SI nie była w stanie „zarobić" na siebie — w USA niewielką grupę ośrodków subsydiowało państwo, w Wielkiej Brytanii taką możliwość przekreślił tzw. raport Lighthilla, który skutecznie przyczynił się do opóźnienia rozwoju SI w tym kraju.

Rzeczywiście, teoretyczny i ezoteryczny nawet charakter początkowych dokonań jawnie kontrastował z oczekiwaniami, jakie opinia publiczna wiązała z „myślącymi maszynami". Liczne zagadnienia okazały się trudniejsze w realizacji niż sądzono początkowo — dobrym tego przykładem był ikarowy lot automatycznego przekładu w późnych latach pięćdziesiątych. Ogólne ujęcia mechanizmu rozwiązywania problemów wypróbowano i doskonalono, rozważając zabawkowe mikroświaty o prostej semantyce, np. świat klocków i zagadnienia związane z przestawianiem elementów w obliczu różnych ograniczeń.

Najwcześniej — bo około 1965 roku — zrozumiano potrzebę konfrontacji technik SI z wymaganiami stawianymi przez rzeczywiste (nie sztuczne, zabawkowe) problemy w uniwersytecie stanfordzkim. W ramach organizacyjnych Heuristic Programming Project, kierowanego przez Edwarda Feigenbauma, wykonano pierwsze systemy ekspertowe: DENDRAL i MYCIN. DENDRAL określa wzór strukturalny związku organicznego na podstawie jego wzoru sumarycznego i spektrogramu masowego, MYCIN diagnozuje choroby zakaźne krwi i opon mózgowych. Oba systemy wyposażono w potrzebną wiedzę specjalistyczną, tak że dorównały swą kompetencją ludziom. Przekonano się, że podobnie jak u człowieka, inteligencja maszynowa najpełniej może wyrazić się w skutecznym manipulowaniu wiedzą — lepiej niż przy rozwiązywaniu sztucznych, abstrakcyjnych problemów. Charakterystyczne jest przy tym, że systemy ekspertowe zawsze stosuje się do wąskich, wysoko wyspecjalizowanych wycinków wiedzy.

Uwaga badaczy przesunęła się z „czystych" mechanizmów wnioskowania na zagadnienia związane z reprezentacją wiedzy i jej wszechstronnego zastosowania. Zwrot ten należy uznać za wyjątkowo szczęśliwy i to co najmniej z trzech powodów. Po pierwsze, dziedzina SI zdobyła sobie niezależny, zewnętrzny „test realności". Po drugie, możliwe stały się zastosowania — a więc poszerzył sią krąg faktycznych i potencjalnych sponsorów. I po trzecie — pojawiły się nowe problemy i nowe rozwiązania, o zasadniczych konsekwencjach dla przyspieszenia rozwoju praktyki i — w mniejszym stopniu — teorii SI.

Systemów ekspertowych wszelkiego rodzaju pojawiła się od 1965 roku bardzo wiele. Nie należy jednak sądzić, że w dziedzinie SI zaniechano badań podstawowych. Dojrzałość do pierwszych zastosowań ma tu niewiele wspólnego z głównymi ambicjami tej dziedziny. Pozostaje jednak faktem, że systemy ekspertowe stanowią obecnie najbardziej obiecujący i najszerzej znany nurt wciąż jeszcze mało popularnej dziedziny SI.

Konsekwencją polityki dotacji jest współczesna „geografia" systemów ekspertowych. Krajem bezwzględnie przodującym w tej dziedzinie są Stany Zjednoczone, gdzie zresztą tylko niewielka liczba ośrodków naprawdę się liczy. Tym niemniej, na jednej z ostatnich międzynarodowych konferencji SI, prace amerykańskie stanowiły ok. 75% ogółu referatów. Na kolejnych miejscach plasują się: Japonia, gdzie technologia ta nabrała wielkiego znaczenia w związku z projektem budowy komputerów piątej generacji, oraz Wielka Brytania, gdzie prace wielu ośrodków koordynuje i finansuje tzw. program Alveya. Poza wymienionymi krajami, rosnące zainteresowanie systemami ekspertowymi można zaobserwować niemal wszędzie. Wśród europejskich krajów

Mgr inż. JACEK AMBROZIAK ukończył w 1984 r. Wydział Elektroniki (Instytut Automatyki) Politechniki Warszawskiej. W czasie ostatniego roku studiów pracował w Instytucie Biocybernetyki i Inżynierii Biomedycznej PAN, a następnie Środowiskowym Centrum Obliczeniowym CYFRONET w Instytucie Energii Atomowej. Obecnie zatrudniony jest w Ośrodku Rozwoju Techniki Instytutu Podstawowych Problemów Techniki PAN.

socjalistycznych Polska zajmuje dalsze miejsce. W szczególności brak jest jakichkolwiek publikacji książkowych poświęconych tej dziedzinie — tłumaczeń czy prac rodzimych. Szeroka fala światowego zainteresowania systemami ekspertowymi zastała nas w stanie głębokiego kryzysu, co w połączeniu z brakiem tradycji w tej dziedzinie daje ów przykry rezultat. Jednakże wielkiej atrakcyjności systemów ekspertowych nie sposób opierać się długo i sądzę, że w ciągu kilku najbliższych lat Polska znajdzie swą drogę w tej obiecującej technice przetwarzania informacji.

CHARAKTERYSTYKA OGÓLNA

Najszerszym kontekstem dla systemów ekspertowych jest ogólna klasa programów SI. Klasę pośrednią stanowią tzw. systemy z bazą wiedzy (ang. knowledge-based systems), w których akcentuje się reprezentacje wiedzy z określonej dziedziny. Systemy ekspertowe stanowią podklasę systemów z bazą wiedzy, z zastrzeżeniem, że mogą spełniać rolę najlepszych specjalistów różnych dziedzin nauki i techniki. Systemy te są bardzo duże i jest ich niewiele. Najliczniejszą grupę systemów z bazą wiedzy stanowią tzw „skill systems", podobne do systemów ekspertowych lecz stawiające sobie o wiele skromniejsze wymagania, a więc mogące zastępować ludzi w zadaniach wymagających wykorzystania wiedzy specjalistycznej, lecz niekoniecznie wiedzy ekspertów. W zastosowaniach systemów ekspertowych można wyróżnić kilka podstawowych obszarów:

- interpretacja danych (sygnałów), tj. przypisywanie im określonego znaczenia (np. DENDRAL lub PROSPECTOR określający położenie złóż metalicznych)
- diagnozowanie źle funkcjonujących systemów (w tym także organizmu ludzkiego, komputerów itp.); do tej klasy należy MYCIN oraz DART/DASD diagnozujący usterki komputerów (dziedziny 1 i 2 często łączy się w ogólniejszą dziedzinę klasyfikacji)
- nadzór — kontrola funkcjonowania złożonych urządzeń technicznych, sprowadzająca się do interpretacji sygnałów pochodzących z tych urządzeń i podejmowania odpowiednich działań (alarm, nastawy parametrów nadzorowanych procesów itp.) w sytuacjach tego wymagających; przykładem jest system VM sprawujący kontrolę nad urządzeniami do sztucznego oddychania przy intensywnej terapii lub system nadzorujący pracę reaktora jądrowego
- przewidywanie przyszłego biegu wypadków na podstawie modelu zjawisk w przeszłości i teraźniejszości
- planowanie, tj. określanie sekwencji czynności doprowadzających do osiągnięcia wyznaczonych celów; przykładem jest zaplanowanie przebiegu syntezy organicznej — SECS, czy eksperymentu genetycznego — MOLGEN
- projektowanie obiektów spełniających określone wymagania, np. układów VLSI.

Najważniejszą, wyróżniającą cechą systemów ekspertowych jest wykorzystywanie specjalistycznej wiedzy z danej dziedziny zastosowania — reprezentowanej w systemie jawnie. Wiedzę reprezentuje się przy użyciu różnych formalizmów (także — kilku równocześnie), ale zawsze systematycznie i w jednolity sposób.

Najczęściej wiedzę reprezentuje się jako zbiór reguł postaci:

JEŻELI ⟨warunki⟩ TO ⟨akcje⟩

W artykule ograniczono się do omówienia tylko tej reprezentacji, jako najpopularniejszej i najprostszej, a przy tym efektywnej.

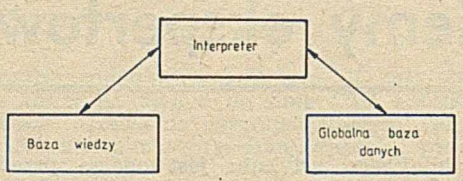
Reprezentacja regułowa charakteryzuje się wieloma użytecznymi cechami — a w szczególności:

- łatwością użycia przez program wnioskujący (tzw. interpreter reguł)
- możliwością wykorzystania reguł do różnych celów (właściwe wnioskowanie, wyjaśnianie wnioskowania, nauczanie)
- modularnością — każda reguła reprezentuje pewien mały, zwarty fragment całej wiedzy systemu
- przejrzystością — regułę można bez trudu opisać w języku naturalnym, jako reprezentującą elementarny pojęciowo krok wnioskowania.
- możliwością przyrostowego tworzenia bazy reguł.

Warunki reguły dotyczą najczęściej cech obiektów rozważanych w danym zadaniu, zaś akcje stanowią wnioski (np. hipotezy) dotyczące tych obiektów (przy klasyfikacji), a także elementy planów (w planowaniu), operacje wejścia-wyjścia (nadzór) itp.

Regułę (tj. związane z nią akcje) wykonuje się, gdy spełnione są jej warunki. Każdorazowy przebieg programu systemu ekspertowego sprowadza się do sekwencji zastosowań (wykonań) pewnego podzbioru reguł z jego bazy wiedzy.

Regułowa reprezentacja wiedzy prowadzi do schematu części wnioskującej (ang. inference engine) systemu ekspertowego, przedstawionego na rys.



Najważniejszą część systemu stanowi baza wiedzy zawierająca reguły. W oczywisty sposób system jest tym lepszy, im lepsze ma reguły. Typowe reguły są zbudowane z kilku — najczęściej od 3 do 5 — warunków, określających stosowalność reguły w danym kontekście i z 1—2 akcji, wyrażających udział reguły w procesie wnioskowania. Liczebność zbioru reguł zależy od kilku czynników: wielkości zadania (w sensie wymaganej wiedzy specjalistycznej), treści reguł (a więc tego, jak drobne reprezentują kroki wnioskowania, w odniesieniu do całego rozważanego problemu) oraz od sposobu ich użycia. Przykładowo, baza wiedzy systemu MYCIN zawiera ok. 450 reguł, system R1 (określający na podstawie wymagań klientów konfiguracje systemów komputerowych VAX) potrzebuje już ok. 800 reguł, a PROSPECTOR — ok. 1800 reguł. Hayes-Roth podaje następujące oszacowanie:

- interesujące zastosowanie możliwości systemu wymaga użycia tylko ok. 50 reguł
- do przekonującego zastosowania z bazą wiedzy potrzeba ok. 250 reguł
- praktyczne, komercjalne systemy wymagają użycia nie mniej niż 500 reguł
- system ekspertowy kompetentny w wąskiej dziedzinie potrzebuje ok. 50—1000 reguł.

Informacje o faktach dotyczących rozwiązywanego problemu umieszcza się w globalnej bazie danych. Jej treść jest dynamiczna — zmienia się w czasie działania programu, a także między jego kolejnymi uruchomieniami. Zawartość bazy reguł pozostaje niezmieniona w tych okolicznościach. Warunki reguł odnoszą się właśnie do elementów globalnej bazy danych, zaś związane z regułami akcje modyfikują jej treść. Ważną cechą systemów regułowych jest to, że reguły nie odnoszą się do siebie nawzajem bezpośrednio, lecz tylko przez modyfikacje globalnej bazy danych — zmiana wywołana przez wykonanie jednej reguły pociąga za sobą możliwość wykonania następnych.

Wykorzystaniem reguł rządzi interpreter. Program ten organizuje sterowanie procesem automatycznego wnioskowania, sprawdza, czy spełnione są w globalnej bazie danych warunki reguł oraz wykonuje akcje skojarzone z regułami. W tym sensie stanowi on moduł wykonawczy systemu.

Algorytm interpretera wyznacza zawsze pewną strategię stosowania reguł, określającą sposób ich użycia i kolejność w jakiej są rozpatrywane. Istnieją dwa zasadnicze reżimy wykorzystania reguł w procesie wnioskowania:

— **wnioskowanie uprzedzające, wnioskowanie do przodu** (ang. data driven, bottom-up, antecedent driven, forward chaining) oraz

— **wnioskowanie wsteczne, wnioskowanie do tyłu** (ang. goal driven, top-down, consequent drivent, backward chaining).

Wnioskowanie uprzedzające

We wnioskowaniu uprzedzającym rozważane są najpierw warunki (przesłanki) reguły. Początkowo, globalna baza danych zawiera opis problemu, np. w systemie R1 (znanym także jako X/Con), są to wymagania klienta dotyczące wa-

runków, które ma spełniać zamawiana przez niego konfiguracja systemu VAX. Interpreter pracuje w cyklu:

1) określenie reguł o przesłaniach spełnionych w globalnej bazie danych

2) wybór jednej spośród nich (ang. conflict resolution)

3) wykonanie akcji wybranej reguły.

Praca interpretera kończy się, gdy zostanie osiągnięty określony warunek końcowy lub gdy w fazie pierwszej zabraknie reguł o spełnionych warunkach. Wówczas globalna baza danych powinna opisywać rozwiązanie problemu. Szczególne znaczenie w technice wnioskowania uprzedzającego mają pierwsze dwie fazy cyklu interpretera.

W pierwszej fazie najważniejsza jest efektywność procesu przeszukującego zbiór reguł. Reguł jest zazwyczaj kilkaset, a więc liczba warunków, które należy sprawdzić, może wynosić kilka tysięcy. Mechanizmy określające spełnienie warunku mogą być również złożone. Cały proces wnioskowania wymaga zazwyczaj wielu pojedynczych cykli interpretera. W tej sytuacji stosuje się różne techniki ograniczania złożoności przeglądania reguł. Pojęciowo najprostsze jest podzielenie zestawu reguł na podzbiory, odpowiadające różnym kontekstom procesu wnioskowania (system R1). W tym rozwiązaniu zawsze tylko fragment bazy wiedzy jest dostępny do przeglądania. W innych technikach wykorzystuje się metareguły rządzące stosowaniem właściwych reguł.

Algorytm wyboru reguły do wykonania (faza 2) jest projektowany tak, aby przy zachowaniu ogólności zapewnić możliwie sensowny przebieg procesu wnioskowania. Jeżeli dla konkretnego problemu istnieje sekwencja zastosowań reguł, która wiedzie do jego rozwiązania, to właśnie druga faza jest odpowiedzialna za jej określenie. Ideałem byłoby wykonywanie tylko reguł potrzebnych w danym wnioskowaniu. W praktyce nie można tego zapewnić dla dużego zbioru reguł i przy zachowaniu względnej prostoty interpretera. Stosuje się więc mechanizmy w miarę nieskomplikowane, lecz oczywiste intuicyjnie. Poniżej przedstawiono listę najczęściej stosowanych kryteriów wyboru reguły:

1) reguły przegląda się w ustalonej kolejności i wybiera pierwszą napotkaną o spełnionych warunkach

2) wybiera się regułę o najwyższym priorytecie, uprzednio nadanym przez programistę

3) do wykonania kieruje się regułę o warunkach wyrażających najbardziej szczegółowy test globalnej bazy danych

4) wybiera się regułę odnoszącą się w swym warunku do danych ostatnio wprowadzonych do bazy

5) wykonuje sie regułę, która dotąd nie była wykonywana w danym kontekście

6) regułę wybiera się arbitralnie.

Oczywiście, konkretny algorytm może łączyć podane kryteria (szczególnie 3—4—5). Jak widać z powyższego przeglądu, projektant systemu ekspertowego musi zadbać o reprezentację wielu faktów — dotyczących samego procesu wnioskowania — nie tylko rozwiązywanego problemu.

Akcje wykonywane w fazie trzeciej mogą mieć różnorodne znaczenie. Oprócz czystego udziału w samym wnioskowaniu, mogą informować użytkownika o przebiegu działania systemu, mogą pytać o potrzebne dane, a także wpływać na parametry, dotyczace mechanizmu procesu wnioskowania i przez to, pośrednio, na wybór podzbioru reguł do rozpatrzenia w pierwszej fazie następnego cyklu.

Wnioskowanie wsteczne

Wnioskowanie wsteczne — zastosowane m.in. w systemach MYCIN i PROSPECTOR — jest wnioskowaniem sterowanym akcjami (konkluzjami) reguł. Zaczyna się od wykonania określonej reguły początkowej, której akcja rozwiązuje cały postawiony problem. Przesłanki tej reguły składają się na podproblemy. Informacje potrzebne przesłankom mogą znajdować się w globalnej bazie danych, system może o nie zapytać użytkownika, wreszcie może je próbować ustalić wykonując inne reguły — te mianowicie, których wnioski dotyczą poszukiwanej informacji. Reguły są więc stosowane ze względu na swoje konkluzje. Interpreter tworzy dynamicznie drzewo reguł, którego korzeniem jest akcja pierwszej reguły, a liśćmi są warunki reguł najbardziej szczegółowych, spełnione przez informacje dostarczone przez użytkownika i (lub) opis faktów z dziedziny, uprzednio umieszczony w bazie wiedzy. W tym reżimie zbiór reguł, stosowalnych na danym etapie, stanowią reguły dostarczające informacji potrzebnych do rozwiązania aktualnego podproblemu. Ponieważ zazwyczaj (np. w programie diagnostycznym) interesuje nas jak najwięcej informacji o podproblemie, stosowalne reguły wykonuje się wszystkie po kolei. Dopiero na etapie sprawdzania warunków reguł, niektóre spośród nich mogą być odrzucone.

Wnioskowanie wsteczne w systemie konsultacyjnym ma trzy ważne zalety:

- Rozwiązywanie kolejnych podproblemów danego problemu narzuca właściwą (z punktu widzenia człowieka) kolejność pytaniom zadawanym użytkownikowi przez system. Pytania nie są chaotyczne, a ogranicza je ścieżka coraz węższych kontekstów jednego, głównego problemu.

- Łatwo jest wyjaśnić użytkownikowi, dlaczego zostało mu postawione pytanie. Wystarczy mianowicie zacytować regułę, której warunek miała spełnić poszukiwana informacja. Pytanie było zadane, aby ustalić zasadność wniosku tej reguły, w kontekście znanych już być może faktów spełniających pozostałe warunki.

- Podobnie — przeglądając drzewo zastosowanych reguł — system może wyjaśnić użytkownikowi, na czym ma się opierać wiara w prawdziwość faktów z globalnej bazy danych. Fakty te mogły bowiem należeć do statycznej wiedzy systemu, mógł je wprowadzić użytkownik w czasie sesji konsultacyjnej, wreszcie system mógł je wydedukować, posługując się swoimi regułami i — rekurencyjnie — innymi faktami.

Ośrodek Postępu Technicznego
NOT

zaprasza do zwiedzenia

STAŁEJ GIEŁDY

ROZWIĄZAŃ TECHNICZNYCH

w Warszawie, ul. Żelazna 51/53
(dawne Zakłady Norblina)

Godziny otwarcia: 9.00 — 15.00

(oprócz sobót i niedziel)

Redakcja
INFORMATYKI
zawiadamia
o stałej sprzedaży
MIKROKLANU
w Klubie Prasy Technicznej
w Warszawie
ul. Mazowiecka 12, tel. 27-43-65

STANISŁAW WRYCZA
Uniwersytet Gdański
Wydział Ekonomiki i Transportu
Sopot

Przesłanki i uwarunkowania modelowania konceptualnego

Modelowanie konceptualne [1]) stało się w ostatnich kilku latach znaczącym kierunkiem w dziedzinie tworzenia i użytkowania baz danych i szerzej — systemów informatycznych [6]. Modelowanie konceptualne stanowi proces tworzenia schematu konceptualnego, czyli zbioru niesprzecznych zdań wyrażających niezbędne twierdzenia o dziedzinie przedmiotowej (badanym wycinku rzeczywistości) [2]. Schemat konceptualny jest zatem sformalizowanym opisem statyki i dynamiki bazy danych reprezentującej wybraną dziedzinę przedmiotową. Najczęściej przyjmuje się opis statyki za pomocą takich pojęć, jak: obiekty, atrybuty i relacje, natomiast dynamiki — za pomocą funkcji zdarzeń i operacji.

Schemat konceptualny spełnia określone role i pewne cechy. Spośród różnych ról schematu warto wymienić:

- sformalizowany model dziedziny przedmiotowej (podstawa do zrozumienia jej zachowania)
- nietechniczny opis bazy danych dziedziny przedmiotowej (podstawa porozumienia się między profesjonalistami-informatykami a użytkownikami bazy danych)
- definicje przekształceń i manipulacji informacją o dziedzinie przedmiotowej.

Powyższe role schematu konceptualnego wyznaczają takie jego cechy, jak niezależność od wszelkich uwarunkowań wynikających ze sposobu przechowywania danych w pamięci, stosowanego systemu zarządzania bazą danych (SZBD), programów i zapytań, a także — stabilność, sformalizowanie i przystępność.

W trakcie tworzenia schematu konceptualnego winny być przestrzegane określone reguły modelowania. Zaproponowano dwie takie podstawowe reguły [2]: regułę 100 procent i regułę konceptualizacji.

Reguła 100 procent:

W schemacie konceptualnym powinny być opisane wszystkie ogólne aspekty dziedziny przedmiotowej.

Reguła konceptualizacji:
Schemat konceptualny powinien obejmować tylko i wyłącznie konceptualnie istotne aspekty dziedziny przedmiotowej, a tym samym wykluczać wszystkie aspekty dotyczące reprezentacji danych, fizycznej organizacji danych i dostępu do danych, jak również wszystkie aspekty dotyczące poszczególnego użytkownika (formaty komunikatów, struktury danych itp.).

Celem niniejszego artykułu jest określenie miejsca i roli modelowania konceptualnego w procesie projektowania baz danych oraz wstępna charakterystyka zaleceń standaryzacyjnych w zakresie tworzenia schematów konceptualnych.

MODELOWANIE KONCEPTUALNE W PROCESIE PROJEKTOWANIA BAZ DANYCH

Proces projektowania baz danych jest przedmiotem wielu opracowań o charakterze metodycznym [4] i porównawczym. Na podstawie ich analizy można stwierdzić, iż występuje duża różnorodność metod projektowania baz danych. Istnieją również pewne podobieństwa między poszczególnymi metodologiami. Wobec wzrostu liczby zastosowań baz danych w różnych dziedzinach działalności, bardzo pożądane jest sformułowanie ujednoliconych reguł projektowania. Najbardziej znaną i uznaną próbę standaryzacji stanowi propozycja grupy 35 autorów, opublikowana w postaci 1978 New Orleans Data Design Workshop Report [3]. Stanowi ona do pewnego stopnia uogólnienie propozycji i zaleceń wypływających z poszczególnych podejść autonomicznych.

Cały proces projektowania bazy danych, przedstawiony na rysunku 1, został podzielony na cztery etapy (obszary projektowe). Granice pomiędzy nimi są do pewnego stopnia rozmyte. Obszar I (Analiza Potrzeb) obejmuje określenie informacyjnych potrzeb przedsiębiorstwa i jego potrzeb w zakresie przetwarzania, bez uwzględnienia struktury przyszłego systemu informacyjnego. Projektowanie w II obszarze dotyczy głównie konstrukcji modelu konceptualnego, niezależnego od konkretnego SZBD. Opis schematu bazy danych nadający się do przetwarzania przy użyciu SZBD stanowi zadanie obszaru III — projektowania wdrożeniowego. Ostatnią fazą jest projekt fizycznej bazy danych, realizowanej przy wykorzystaniu dostępnego bądź wybranego sprzętu informatycznego i oprogramowania. Należy podkreślić, że w całym cyklu tworzenia baz danych modelowanie konceptualne zajmuje miejsce centralne. Niemniej istotne znaczenie odgrywa ta faza także w innych, nie omawianych tu metodykach projektowania baz danych.

Wejściem do fazy modelowania konceptualnego (obszar II) są wyniki prac wykonywanych w pierwszym etapie projektowania, w postaci dokumentów, wypełnionych kwestionariuszy, diagramów i opisów wykonanych w językach sformalizowanych. Główne zadania realizowane w obrębie obszaru II są następujące:

— analiza i modelowanie informacyjnych potrzeb i ograniczeń przedsiębiorstwa oraz użytkowników na podstawie odpowiedniego modelu konceptualnego, w celu wytworzenia pożądanych modeli zastosowań i modeli kierowniczych
— analiza i modelowanie potrzeb w zakresie przetwarzania
— integracja modeli zastosowań i modeli kierowniczych we wspólny model przedsiębiorstwa, metodą wykrycia i usunięcia sprzeczności jak również modyfikacji modeli wejściowych.

Rezultatami czynności projektowych wykonywanych w obszarze II wprowadzonymi do obszaru III są: model konceptualny bazy danych, modyfikowane modele zastosowań, niesprzeczne z modelem konceptualnym, oraz potrzeby dostępu do bazy danych ujęte w kategoriach modelu konceptualnego.

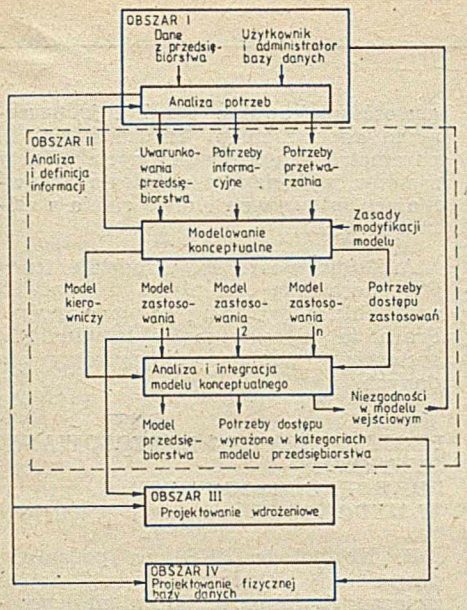
ZAŁOŻENIA STANDARYZACYJNE W ZAKRESIE TWORZENIA SCHEMATÓW KONCEPTUALNYCH

Kwestia tworzenia schematów konceptualnych była przedmiotem prac i zaleceń standaryzacyjnych. Najbardziej znane zalecenia zostały zaproponowane przez ANSI/X3/SPARC [2]) [5] oraz ISO TC97/SC5/WG3 [3]) [2]. Głównym

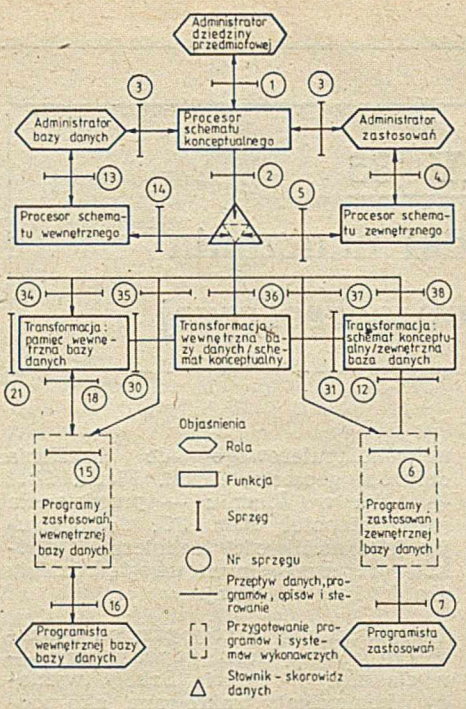
[1]) Zwane też modelowaniem pojęciowym (przyp. red.)

[2]) ANSI/X3/SPARC — American National Standards Institute Study Group on Database Management Systems (przyp. aut.)

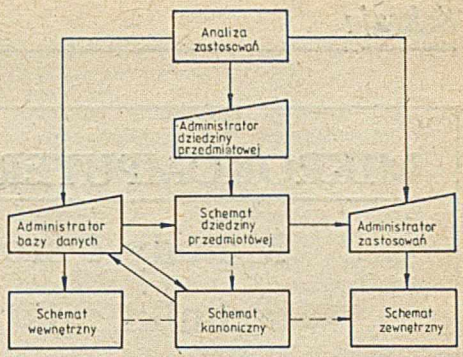
[3]) ISO/TC97/SC5/WG3 — International Organization for Standardization Working Group on the Aspects of Conceptual Schemata for Data Base Management Systems (przyp. aut.)



Rys. 1. Uogólniony proces projektowania bazy danych [3]



Rys. 2. Struktura systemu zarządzania bazą danych według propozycji ANSI/X3/SPARC (wycinek) [5]



Rys. 3. Czteropoziomowa architektura systemu zarządzania bazą danych [1]

punktem propozycji ANSI była koncepcja trójpoziomowego standardu SZBD, którego wycinek przedstawia rys. 2. Pełny opis systemu zarządzania bazą danych zamieszczony m.in. w [5], jest znacznie bardziej rozbudowany w części dotyczącej schematu wewnętrznego. Obejmuje więc m.in. takie aspekty, jak: podział pamięci, wpływ urządzeń komputera na integralność, niezależność i bezpieczeństwo danych, udział podsystemów zapytań, aktualizacji czy generowania zestawień, przesyłanie danych na urządzenia pamięci itd.

Głównym celem prac grupy badawczej ANSI/X3/SPARC było określenie zestawu wymagań dla efektywnych SZBD. Zasadniczą cechą omawianej propozycji standaryzacji jest oparcie struktury SZBD na ściśle wyspecyfikowanych i zdefiniowanych sprzęgach (ang. interfaces), czyli określenie jak poszczególne części systemu są zazębione, sprzężone, a nie — jak części te mają funkcjonować. Pełną listę sprzęgów zawiera m.in. praca [5].

Proponowane przez grupę badawczą ANSI/X3/SPARC trzy poziomy SZBD, a także odpowiadające im schematy noszą nazwy: **konceptualny** (ang. conceptual), **zewnętrzny** (ang. external) i **wewnętrzny** (ang. internal). Schemat zewnętrzny uwzględnia wymagania zastosowań (perspektywa programisty) bazy danych, a schemat wewnętrzny — systemu komputerowego (perspektywa systemu). Schemat konceptualny jest sformalizowanym opisem badanej rzeczywistości, zawierającym definicje obiektów, atrybutów i relacji.

Spośród przedstawionych na rys. 2 składników systemu warto zwrócić uwagę na role związane z administrowaniem oraz zadanie słownika—skorowidza danych. Autorzy opracowania zalecają trzy główne role w zakresie administrowania bazą danych (por. rys. 2): **rolę administratora dziedziny przedmiotowej** (ang. enterprise administrator), **administratora zastosowań** (ang. application system administrator) i **administratora bazy danych** (ang. database administrator). Należy zaznaczyć, że każda z ról może być spełniona niekoniecznie przez pojedynczą osobę, lecz przez grupę osób. Cele realizowane przez poszczególnych administratorów dotyczą przede wszystkim tworzenia i eksploatowania schematów, odpowiednio — konceptualnego, zewnętrznego i wewnętrznego. W słowniku-skorowidzu przechowywane są wszystkie definicje i reguły związane ze schematami i transformacjami między nimi.

Zasadnicze rozwiązania, przedstawione wyżej, zostały przyjęte w raporcie grupy badawczej ISO [2]. Jednak w tym opracowaniu problemy modelowania konceptualnego zostały potraktowane szerzej i odniesione ogólnie do systemów informacyjnych. Grupa zdefiniowała szereg pojęć, opracowała reguły konstruowania schematu konceptualnego, dokonała porównań trzech głównych podejść do modelowania konceptualnego [4], zaproponowała odpowiadającą im strukturę języków opisu schematu konceptualnego. Praca ta stanowi istotny wkład do teorii i praktyki modelowania konceptualnego.

Struktura SZBD, przedstawiona w [5], jest wciąż doskonalona. Ważną propozycję w tym zakresie stanowi rozszerzony czteroschematowy model architektury systemu bazy danych De, Hasemana i So [1] (rys. 3). Dokonano w nim podziału schematu konceptualnego na **schemat dziedziny przedmiotowej** (ang. enterprise schema) i **schemat kanoniczny** (ang. canonical schema). Schemat dziedziny przedmiotowej jest „czystym" opisem wycinka rzeczywistości, niezależnym od uwarunkowań pamięci i efektywności, konceptualnym opisem zawartości bazy danych. Schemat kanoniczny stanowi realizację schematu dziedziny przedmiotowej w kategoriach struktur danych. Rozróżnienie wiąże się z oddzieleniem dwu istotnych zadań schematu konceptualnego — semantycznego opisu wiążącego bazę danych z dziedziną przedmiotową oraz opisu składniowego, działającego jako sprzęg między użytkownikiem (schemat zewnętrzny) a pamięcią (schemat wewnętrzny). Architektura SZBD zaprezentowana na rys. 3, choć nie uzyskała dotąd szerszego uznania, w znacznie większym zakresie uwzględnia rozwój teorii modelowania konceptualnego, a zwłaszcza języków opisu schematu konceptualnego.

[4] Będzie o nich mowa w następnym artykule (przyp. red.)

LITERATURA

[1] De P., Haseman W. D., So R. H.: Four Schema Approach — An Extended Model for Data Base Architecture. Information Systems, Vol. 6, No 2, pp. 117—124, 1981
[2] Griethuysen J. J. et al. (eds.): Concepts and Terminology for the Conceptual Schema and Information Base. ISO TC97/SC5/WG3, February 1982
[3] Lum V. Y. et al.: 1978 New Orleans Data Base Design Workshop Report. Pp. 328—339, Proc. Fifth Intern. Conf. on Very Large Data Bases, Rio de Janeiro, October 3—5, 1979
[4] Oleński J., Staniszkis W.: Projektowanie bazy danych. PWE, Warszawa, 1984
[5] Tsichritzis D., Klug A.: ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems. Information Systems, Vol. 3, pp. 173—191, 1978
[6] Wrycza S.: Aktualne trendy w modelowaniu konceptualnym. Materiały Szkoły Baz Danych PTI, Karpacz 1985, str. 31—65.

MIKROKOMPUTERY '85

Metody i środki nauczania techniki mikroprocesorowej

Instytut Cybernetyki Technicznej Politechniki Wrocławskiej zorganizował I Międzynarodową Szkołę nt. „MIKROKOMPUTERY — PROJEKTOWANIE — PRAKTYKA — NAUCZANIE" [1]), która odbyła się w Bierutowicach k. Karpacza w dniach 24—27 września 1985 r.

Celem Szkoły była wymiana doświadczeń z zakresu metod i środków nauczania techniki mikroprocesorowej w szkołach wyższych oraz w ramach szkoleń specjalistów z przemysłu (studia podyplomowe, doskonalenie kadr inżynierskich itp.). Tematyka Szkoły obejmowała następujące zagadnienia:

- podstawy techniki mikroprocesorowej,
- projektowanie i zastosowanie systemów mikroprocesorowych w dydaktyce,
- nauczanie techniki mikroprocesorowej,
- organizacja laboratoriów dydaktycznych.

Obradom towarzyszyła wystawa sprzętu mikrokomputerowego produkowanego w Polsce. Instytut Cybernetyki Technicznej Politechniki Wrocławskiej zaprezentował mikrokomputery laboratoryjne z zastosowaniem mikroprocesorów Z-80 i INTEL 8080. Przedstawiono zestaw stanowisk laboratoryjnych wykorzystywanych w Laboratorium Systemów Mikroprocesorowych Instytutu przez studentów IV roku Wydziału Elektroniki o specjalności systemy informatyczne na zajęciach z tematów: port we-wy, sterowanie transmisją szeregową, tester układów logicznych TTL, model skrzyżowania, rejestracja informacji cyfrowej w standardzie KANSAS CITY, programowe sterowanie silnika prądu stałego. Przedsiębiorstwo Zagraniczne AMEPROD z Poznania przedstawiło zestaw laboratoryjny ZLA-01 przeznaczony do ćwiczeń dydaktycznych z dziedziny techniki mikroprocesorowej i oprogramowania. Instytut Fizyki Uniwersytetu Gdańskiego wystawił klawiaturę do mikrokomputera SINCLAIR ZX-81 oraz interfejs sprzęgający ZX-81 z woltomierzem cyfrowym i silnikiem krokowym wraz z oprogramowaniem (ASSEMBLER, Basic). Spółdzielnia Pracy Mikroprocesorowych Systemów Automatyki MIKRONIKA z Poznania zaprezentowała zastosowanie systemu mikrokomputerowego do kreślenia obrazów geometrycznych. Firma IBM — filia w Wiedniu — przedstawiła mikrokomputer IBM PC, demonstrując jego możliwości techniczne i użytkowe ze szczególnym uwzględnieniem grafiki komputerowej.

Organizatorzy Szkoły przyjęli dyskusję za główną formę dydaktyczną — wygłaszane w trakcie Szkoły wykłady i komunikaty traktowano jako podstawę do dyskusji.

PODSTAWY TECHNIKI MIKROPROCESOROWEJ

R. Jacewicz i J. Majewski zaprezentowali uwagi związane z organizacją systemów operacyjnych typu monitora. W celu zwiększenia ich elastyczności i efektywności zaproponowali usprawnienia polegające na możliwości:

- wywoływania komend z programu użytkownika,
- rozbudowy systemu o dodatkowe komendy.

Pracę zilustrowali wydrukiem pętli głównej systemu operacyjnego przeznaczonego dla mikroprocesorów INTEL 8080/85 i Z-80.

I. Jelonek, J. R. Just, M. Gołoś przedstawili ogólną ideę realizacji rozproszonego systemu komputerowego, składającego się z trzech mikrokomputerów, przy założeniu, że mikrokomputery te komunikują się ze sobą przez pamięci buforowe — za pomocą operacji wejścia-wyjścia.

B. Michalak opisał system wspomagania projektowania programów w języku asemblera mikroprocesora INTEL 8080. System jest przeznaczony dla mikrokomputerów wyposażonych w sekwencyjne nośniki informacji zewnętrznej (taśmy magnetyczne lub papierowe) i umożliwia:

- asemblację w trybie konwersacyjnym,
- relokację oraz modyfikację kodu wynikowego w pamięci operacyjnej,
- konwersacyjną modyfikację etykiet i komentarzy.

J. Piecha przedstawił 32-bitowy system mikroprocesorowy HP 9000 oparty na elementach o bardzo dużym stopniu scalenia (w technologii VLSI), składający się z pięciu bloków zintegrowanych: jednostki centralnego procesora (CPU), procesora wejściowo-wyjściowego, bloku zegarowego, sterownika bloku pamięci oraz bloku RAM. Projekt systemu powstał na podstawie optymalnej technologii NMOS-III, zapewniającej bardzo dużą gęstość upakowania VLSI z jednoczesnym zapewnieniem bardzo dobrych parametrów eksploatacyjnych. Karty systemu wykonano technologią zapewniającą intensywną emisję ciepła z elementu VLSI.

A. Dubielewicz i in. przedstawili nową, sterowaną przepływami danych architekturę systemu komputerowego. Praca ta wprowadza w terminologię systemów sterowanych przepływem danych.

PROJEKTOWANIE I ZASTOSOWANIE SYSTEMÓW MIKROPROCESOROWYCH W DYDAKTYCE

U. Kosmol (z NRD) na podstawie programu niektórych przedmiotów związanych z wykształceniem przyszłych projektantów sprzętu komputerowego ustalił warunki, którym powinien odpowiadać sprzęt mikroprocesorowy używany w laboratoriach wyższych uczelni. Dokonał porównania mikrokomputerów dostępnych na rynkach Polski i NRD oraz ocenił przydatność oferowanych modeli do użytku w procesie kształcenia projektantów sprzętu komputerowego.

P. Misiurewicz omówił zasadnicze zagadnienia, które są przedstawiane na wykładach na temat zastosowania mikroprocesorów w układach pomiarowych i sterujących. Zagadnienia te są związane ze sprzęganiem mikroprocesora z nietypowymi urządzeniami zewnętrznymi oraz ze specyfiką pracy mikroprocesora sterującego w czasie rzeczywistym. Specyfika ta obejmuje takie problemy jak realizacja współbieżności, synchronizacja, wymiana danych pomiędzy współbieżnymi programami, specyficzne algorytmy obliczeniowe dla mikroprocesorów.

H. B. Bemmann (z NRD) przedstawił podstawy projektowe samotestowania systemu mikrokomputerowego. Przeprowadził przegląd stosowanych technik testowania. Opisał prosty sposób generacji sekwencji samotestujących oraz podał algorytm samotestowania systemów mikrokomputerowych.

T. Nowak i J. Haraźny przedstawili konstrukcję i oprogramowanie modułowego systemu mikrokomputerowego USM 6800 oraz możliwość jego zastosowania w procesie dydaktycznym lub w układach sterowania. Zaproponowany system ma wielopoziomową strukturę oprogramowania oraz modularną konstrukcję, co umożliwia wykorzystanie go do wielu różnych zadań, począwszy od nauczania architektury najprostszego systemu, a skończywszy na złożonym układzie z rozbudowanym systemem operacyjnym.

W. Barański i R. Kołodziej przedstawili właściwości, możliwości zastosowań, budowę oraz oprogramowanie monitora graficznego o rozdzielczości 256 × 256 punktów. Podstawową dziedziną zastosowań urządzenia jest zobrazowanie satelitarnych danych meteorologicznych, przesyłanych kanałami o zmniejszonej zdolności rozdzielczej (WEFAX, APT).

[1]) Mikrokomputery '85 — projektowanie, praktyka, nauczanie. Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej nr 69, seria: Konferencje nr 28, Wydawnictwo Politechniki Wrocławskiej, Wrocław 1985.

I. Jóźwiak i J. Machowski zaproponowali koncepcję budowy węzła sieci komputerowej z zastosowaniem mikroprocesora INTEL 8085. Zaproponowany węzeł jest przeznaczony do pracy w sieciach z komutacją pakietów, a wymiana informacji między węzłami sieci następuje na podstawie protokołu BSC.

W. Figiel przedstawił metodę modelowania mikroprocesorowej struktury sterującej przez ograniczoną deterministyczną sieć Petriego. Przedstawił również metodę badania efektywności systemu sterowania.

NAUCZANIE TECHNIKI MIKROPROCESOROWEJ

Na ten temat przedstawiono 7 referatów, przy czym nie wszystkie zostały opublikowane. Spośród nieopublikowanych bardzo duże zainteresowanie wzbudził referat T. Golonki z Akademii Rolniczej z Krakowa nt. *„Mikrokomputer osobisty narzędziem wspomagającym wykładowcę w procesie dydaktycznym"*.

Ponadto M. Nycz i U. Mleczko z Akademii Ekonomicznej we Wrocławiu poprowadziły wykłady nt. *„Mikrokomputer osobisty narzędziem wspomagającym wykładowcę w procesie dydaktycznym"* oraz *„Znaczenie rysunku, barwy i dźwięku w procesie nauczania wspomaganego komputerem"*.

M. Barton (z Wielkiej Brytanii) omówił sposoby nauczania techniki mikrokomputerowej w ramach samodzielnych zajęć laboratoryjnych i grupowych zajęć projektowych na Uniwersytecie Bristolskim. Zaprezentował także uwagi krytyczne o skuteczności zastosowanych metod nauczania.

R. Jacewicz i J. Majewski przedstawili jednopłytowy komputer dydaktyczny zbudowany w Instytucie Cybernetyki Technicznej Politechniki Wrocławskiej, przeznaczony do nauczania organizacji sprzętu mikrokomputerowego oraz programowania w języku wewnętrznym. Rozbudowa systemu wejścia-wyjścia umożliwia mikrokomputerowi współpracę z wieloma urządzeniami zewnętrznymi, takimi jak programatory, testery itd.

H. Stelmasik i J. Zabrodzki zaproponowali koncepcję programu studium podyplomowego „Systemy mikroprocesorowe". Omówili problemy związane z przygotowaniem oraz realizacją studium podyplomowego w kontekście dziennych studiów magisterskich. Zaproponowali szczegółowy program oraz wnioski płynące z realizacji studium przez Instytut Informatyki Politechniki Warszawskiej w latach 1983/1984 i 1984/1985.

K. Jaworek opracował czterosemestralny kurs nauczania techniki mikroprocesorowej dla wybranej kilkunastoosobowej grupy studentów ze specjalności mechanika stosowana na Wydziale Mechanicznym Politechniki Warszawskiej. W artykule zaprezentował program nauczania obejmujący: budowę mikroprocesorów, algorytmy arytmetyki maszyn cyfrowych, programowanie mikroprocesorów w języku wewnętrznym oraz program ćwiczeń praktycznych na mikrokomputerach w specjalnie do tego celu zorganizowanym laboratorium programowania mikroprocesorów.

ORGANIZACJA LABORATORIÓW DYDAKTYCZNYCH

Na ten temat wygłoszono 7 referatów. Szczególne zainteresowanie wzbudziło wystąpienie D. Cabana, który przedstawił koncepcję laboratorium systemów mikroprocesorowych, tworzonego w Instytucie Cybernetyki Technicznej Politechniki Wrocławskiej w ramach specjalności systemy informatyczne. Scharakteryzował założenia dydaktyczne, sprzęt oraz oprogramowanie laboratorium (stanowiska laboratoryjne są realizowane w postaci modułowej z niektórymi modułami specjalizowanymi do celów nauczania).

Ponadto V. Dvorak (z CSRS) zaprezentował program nauczania projektowania i wykorzystywania systemów mikroprocesorowych na Wydziale Elektrycznym Politechniki w Brnie. Na podstawie doświadczeń dydaktycznych przedstawił analizę wykorzystania jednopłytowych mikrokomputerów na zajęciach laboratoryjnych.

M. Chałon omówiła charakterystykę mikrokomputerów serii ELWRO 500 produkowanych w ZE ELWRO we Wrocławiu. Podała możliwości ich wykorzystania w laboratoriach dydaktycznych.

Z. Banaszak, M. Mazur opisali budowę i działanie samoprogramującego się sterownika procesów sekwencyjnych opartego na mikroprocesorze Z-80. Prezentowane rozwiązanie sterownika jest podstawowym elementem stanowiska laboratoryjnego przeznaczonego do elastycznego programowania linii produkcyjnych.

A. Skorupski przedstawił propozycję wyposażenia laboratorium studenckiego do nauczania techniki mikroprocesorowej w Instytucie Informatyki Politechniki Warszawskiej. Opisał przykładowe urządzenia cyfrowe będące wyposażeniem tego laboratorium.

J. Mirkowski, A. Piątkowski, M. Kosicka przedstawili koncepcję sieci laboratorium dydaktycznego, opartego na rodzinie minikomputerów PDP-11. Organizacja laboratorium pozwala na realizację ćwiczeń i eksperymentów badawczych w zakresie fizyki i elektroniki jądrowej i medycznej oraz na wykorzystanie całości sieci w charakterze ośrodka obliczeniowego.

R. Świniarski omówił problemy organizacji laboratorium dydaktycznego i badawczego w zakresie mikrokomputerowych systemów sterowania w czasie rzeczywistym, zrealizowanego w Instytucie Sterowania i Elektroniki Przemysłowej Politechniki Warszawskiej. Opisał sprzęt i oprogramowanie podstawowe oraz koncepcję ćwiczeń projektowych w zakresie systemów mikrokomputerowych do pomiarów i sterowania.

* * *

Szkoła MIKROKOMPUTERY '85 cieszyła się dużym zainteresowaniem, co wskazuje na potrzebę organizowania dalszych konferencji tego rodzaju. Mając na uwadze duże zainteresowanie techniką mikroprocesorową, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej zamierza organizować corocznie Międzynarodową Szkołę pod nazwą: MICROCOMPUTERS — DESIGN — PRACTICE — EDUCATION. Tematy kolejnych edycji będą się zmieniać zgodnie ze światowymi tendencjami w problematyce techniki mikroprocesorowej. Planowanym tematem na rok 1986 jest niezawodność i diagnostyka systemów mikroprocesorowych.

Organizatorzy mają nadzieję, że Szkoły zintegrują środowisko naukowo-badawcze zajmujące się techniką mikroprocesorową i będą doskonałym forum wymiany doświadczeń w tej dziedzinie.

Dr inż. IRENEUSZ J. JÓŹWIAK

Ceny ogłoszeń

Od 1 stycznia br. obowiązują następujące ceny ogłoszeń publikowanych na naszych łamach:
ogłoszenia duże (zależnie od objętości):
cała strona — 35 tys. zł, 3/4 — 30 tys., 1/2 — 25 tys., 1/4 — 20 tys., 1/8 — 15 tys.

ogłoszenia drobne (zależnie od liczby słów):
jedno słowo — 30 zł

Dodatki do ceny podstawowej:
— za dodatkowy kolor (na okładce) +30%
— za zamieszczenie ogłoszenia na czwartej stronie okładki +100%
— za zamieszczenie ogłoszenia na trzeciej stronie okładki +50%

Zniżki:
— za ogłoszenie 3—5-krotne —5%
— za ogłoszenia 6—10-krotne —10%
— za ogłoszenia 11-krotne i powyżej —20%
— za artykuły reklamowe i wkładki wykonane przez zleceniodawcę —40%
— za bloki i biuletyny wykonane przez zleceniodawcę — maks. —60%

Kto jest kim w IFIP

Robert Piloty (RFN)

Na Zgromadzeniu Ogólnym IFIP, we wrześniu 1984 roku w Warnie, profesor Robert Piloty zakończył swoją drugą kadencję jako wiceprezydent, którym był od 1980 roku. Chociaż opuścił Radę IFIP, pracuje nadal dla tej organizacji jako przedstawiciel RFN w Zgromadzeniu Ogólnym i zastępca przewodniczącego Komitetu Technicznego Projektowania Systemów Cyfrowych (Technical Committee No. 10).

Prof. Piloty rozpoczął edukację w Wyższej Szkole Technicznej w Monachium, gdzie w 1949 roku otrzymał tytuł doktora nauk technicznych. Karierę zawodową rozpoczął w Institut für Nachrichtentechnik, ucząc i prowadząc badania w dziedzinie teorii informacji i falowodów. Jego kontakt z komputerami zaczął się od projektu PERM, w którym był dyrektorem technicznym ds. konstrukcji jednego z pierwszych komputerów elektronicznych w Niemczech. Pracując w przemyśle, kierował również planowaniem i wytwarzaniem pierwszych elektronicznych systemów rezerwacji miejsc w liniach lotniczych.

Obecnie R. Piloty jest profesorem elektroniki i informatyki w Wyższej Szkole Technicznej w Darmstadt. Stanowisko to piastuje od 1964 roku. Uczy projektowania systemów cyfrowych i architektury komputerów oraz przewodniczy kilku rządowym komitetom wprowadzających informatykę do kształcenia uniwersyteckiego. Dodatkowo prowadzi badania nad możliwością stosowania projektowania wspomaganego komputerowo (CAD) w procesie projektowania komputerów. Jest przewodniczącym międzynarodowej grupy roboczej, która stworzyła przeznaczony do projektowania komputerów język CONLAD. Jest także autorem dwóch książek i wielu artykułów w czasopismach technicznych.

Jego praca w IFIP rozpoczęła się w 1976 roku, kiedy został wybrany do Zgromadzenia Ogólnego. R. Piloty przyczynił się do utworzenia i rozwoju Komitetu TC 10 najpierw będąc przewodniczącym grupy roboczej IFIP w tej dziedzinie, a następnie przewodniczącym TC 10 od 1976 do 1980 roku. Do Rady IFIP został wybrany w 1978 roku, a jej wiceprezydentem został w 1980 roku.

Profesor Piloty jest od dzieciństwa zapalonym narciarzem, gra również w tenisa. Jest żonaty od 35 lat i ma dwoje dzieci.

(MK)

KONFERENCJE

EUROCAL'87
Europejska Konferencja Algebry Komputerowej

Doroczna konferencja poświęcona obliczeniom symbolicznym i algebraicznym odbędzie się w Lipsku, w dniach 2—5 czerwca 1987 roku. Jest organizowana pod patronatem SAME (Symbolic and Algebraic Manipulation in Europe) i przy współudziale ACM SIGSAM.

Tematyka konferencji obejmuje następujące dziedziny:

- **algorytmy symboliczne, algebraiczne i analityczne,**
- **automatyczne dowodzenie twierdzeń i automatyczne programowanie,**
- **geometria obliczeniowa,**
- **języki i systemy obliczeń symbolicznych,**
- **zastosowania obliczeń symbolicznych w nauce, technice i edukacji.**

Przewidziano trzy rodzaje prezentacji: pełne artykuły badawcze (12 stron), artykuły skrócone (2 strony) oraz sesje plakatowe. Materiały konferencyjne z wyjątkiem prezentacji plakatowych zostaną opublikowane przez Wydawnictwo Springer-Verlag. Komitet Organizacyjny oczekuje propozycji referatów przeglądowych oraz zachęca do demonstracji programów i sprzętu komputerowego.

Przewodniczącym Konferencji i Komitetu Organizacyjnego jest Wolfgang Lassner, do którego należy pisać pod adresem:
Wolfgang Lassner
Mathematics Section
Karl-Marx-University, Leipzig 7010, NRD

Artykuły (w czterech egzemplarzach) należy natomiast przesyłać pod adresem: przewodniczącego Komitetu Programowego:
J. H. Davenport
School of Mathematical Sciences
University of Bath
Bath BA2 7AY, Wielka Brytania
w terminie do 15 stycznia 1987 r.

Informacje o przyjęciu artykułów zostaną rozesłane do 1 kwietnia 1987 r.

(M.M.)

„Zastosowanie mikrokomputerów w ośrodkach inte"

Sympozjum nt. zastosowania mikrokomputerów w ośrodkach informacji naukowej, technicznej i ekonomicznej będzie pierwszym spotkaniem w Polsce. Celem zasadniczym, jaki przyświeca konferencji, jest ocena stanu naszej wiedzy i osiągnięć praktycznych w tej dziedzinie. Sympozjum ma służyć wymianie doświadczeń i informacji. W ramach sympozjum przewiduje się następujące bloki tematyczne:

- **dziedziny inte, w których mogą znaleźć zastosowanie mikrokomputery,**
- **języki informacyjno-wyszukiwawcze,**
- **oprogramowanie do celów inte,**
- **dane techniczne i eksploatacyjne dostępnych na rynku mikrokomputerów.**

Imprezą towarzyszącą sympozjum będzie wystawa sprzętu komputerowego i urządzeń pomocniczych oraz prezentacja programów. Uczestnicy będą mieli okazję zwiedzić ekspozycję XXIII Międzynarodowej Wystawy Literatury Firmowej.

Termin i miejsce imprezy: 18—20 listopada 1987 r., Ośrodek Postępu Technicznego, ul. M. Buczka 1b, Katowice.

Organizatorami sympozjum są:
Instytut Metali Nieżelaznych w Gliwicach, Ośrodek Postępu Technicznego w Katowicach oraz Komisja INTE Stowarzyszenia Inżynierów i Techników Przemysłu Hutniczego, Oddział w Katowicach. Patronal nad imprezą objął Urząd Postępu Naukowo-Technicznego i Wdrożeń.

W. Młodożeniec

Cellary W.: Systemy wielomikroprocesorowe (1)

INFORMATYKA 1986, nr 11—12, s. 1

Pierwsza część charakterystyki systemów wielomikroprocesorowych, zawierająca omówienie architektury systemów jednomikroprocesorowych oraz architektury wewnętrznej mikroprocesora Intel 8086.

Целлары В.: Многомикропроцессорные системы (1)

INFORMATYKA 1986, № 11—12, с. 1

Первая часть характеристики многомикропроцессорных систем, содержащая описание архитектуры одномикропроцессорных систем и внутренней архитектуры микропроцессора Интел 8086.

Zakharov V.: Lokalna sieć komputerowa ETHERNET (2)

INFORMATYKA 1986, nr 11—12, s. 4

Druga część charakterystyki sieci ETHERNET, zawierająca omówienie sposobu przyłączania do tej sieci urządzeń i komputerów, a także przykładową jej realizację na Uniwersytecie Genewskim.

Захаров В.: Локальная компьютерная сеть ETHERNET (2)

INFORMATYKA 1986, № 11—12, с. 4

Вторая часть характеристики сети ETHERNET, содержащая описание способа подключения к этой сети устройств и компьютеров, а также ее примерную реализацию на Женевском Университете.

Zieliński K., Indulska J., Krasowski R.: Lokalna sieć komputerowa UMMLAN-2

INFORMATYKA 1986, nr 11—12, s. 8

Definicja pojęcia sieci lokalnej, charakterystyka aktualnego stanu prac normalizacyjnych oraz rozwoju technologii VLSI dla tego typu sieci, a także omówienie architektury sprzętowej i oprogramowania sieci UMMLAN-2, zrealizowanej w Akademii Górniczo-Hutniczej w Krakowie.

Зелинский К., Индульская Я., Красовский Р.: Локальная компьютерная сеть UMMLAN-2

INFORMATYYKA 1986, № 11—12, с. 8

Определение понятия локальной сети, характеристика настоящего состояния работ в области стандартизации и развития технологии для сетей такого типа, а также описание архитектуры оборудования и программного обеспечения сети UMMLAN-2, реализованной в краковской Горно-металлургической академии.

Dworakowski W.: Komputery osobiste IBM PC (3)

INFORMATYKA 1986, nr 11—12, s. 11

Trzecia część charakterystyki sprzętowej komputerów IBM PC, zawierająca omówienie kolejnych, dostępnych na rynku krajowym, sterowników urządzeń zewnętrznych.

Двораковский В.: Персональные компьютеры IBM PC (3)

INFORMATYKA 1986, № 11—12, с. 11

Третья часть характеристики оборудования компьютеров IBM PC, содержащая описание очередных, доступных на отечественном рынке, контроллеров периферийных устройств.

Grabowicz R., Zalewski J.: System operacyjny PC-DOS (3)

INFORMATYKA 1986, nr 11—12, s. 13

Dokończenie charakterystyki systemu operacyjnego PC-DOS, zawierające omówienie pozostałych rodzajów poleceń tego systemu.

Грабович Р., Залевски Я.: Операционная система PC-ДОС (3)

INFORMATYKA 1986, № 11—12, с. 13

Завершение характеристики операционной системы PC-ДОС содержащее описание остальных видов команд этой системы.

Maćków P.: Komunikacja szeregowa w mikrokomputerach IBM PC i kompatybilnych

INFORMATYKA 1986, nr 11—12, s. 15

Omówienie problemu wykorzystania i współdziałania istniejącego sprzętu komputerowego do współpracy z mikrokomputerami IBM PC i kompatybilnymi przez wykorzystanie złącza szeregowego. Omówiono strukturę sprzętową takiego złącza oraz oprogramowanie kanału szeregowego.

Мацьков П.: Последовательная связь в микроЭВМ IBM PC и совместимых

INFORMATYKA 1986, № 11—12, с. 15

Описание вопроса использования и взаимодействия существующего вычислительного оборудования с микроЭВМ IBM PC и совместимыми путем использования последовательного соединения. Описание структуры оборудования такого соединения и программное обеспечение последовательного канала.

Ambroziak J.: Systemy ekspertowe (1)

INFORMATYKA 1986, nr 11—12, s. 19

Pierwsza część charakterystyki systemów ekspertowych, zawierająca omówienie genezy tego rodzaju systemów oraz ich typowych rozwiązań.

Амброзяк Я.: Экспертные системы (1)

INFORMATYKA 1986, № 11—12, с. 19

Первая часть характеристики экспертных систем, содержащая описание генезиса систем такого вида и их типовых решений.

Wrycza S.: Przesłanki i uwarunkowania modelowania konceptualnego

INFORMATYKA 1986, nr 11—12, s. 22

Charakterystyka istoty modelowania konceptualnego i jego zastosowania w procesie projektowania baz danych. Omówiono tematykę prac normalizacyjnych prowadzonych w tej dziedzinie przez ANSI oraz ISO.

Врыча С.: Основные положений и обусловленности концептуального моделирования

INFORMATYKA 1986, № 11—12, с. 22

Характеристика сущности концептуального моделирования и его применения в процессе проектирования баз данных. Описание тематики работ по стандартизации, проводимых в этой области ANSI и ISO.

Cellary W.: Multimicroprocessor systems (1)

INFORMATYKA 1986, No. 11—12, p. 1

First part of the characteristics of the multimicroprocessor systems, which includes discussion of the architecture of single microprocessor systems and internal description of the Intel 8086 chip.

Cellary W.: Mehrmikroprozessorsysteme (1)

INFORMATYKA 1986, Nr. 11—12, S. 1

Erster Teil einer Charakteristik von Mehrmikroprozessorsystemen, der eine Besprechung der Einmikroprozessorsystemearchitektur, sowie der Intel 8086-Innenarchitektur, umfasst.

Zakharov V.: The ETHERNET — a local computer network (2)

INFORMATYKA 1986, No. 11—12, p. 4

Second part of the characteristics of the ETHERNET network, which includes discussion of the device and computer connections to this network, as well as the description of its implementation in the Geneva University.

Zakharov V.: Lokales Computernetz ETHERNET (2)

INFORMATYKA 1986, Nr. 11—12, S. 4

Zweiter Teil einer Charakteristik des ETHERNET-Computernetzes, der eine Besprechung des Anschlusses von Einrichtungen und Computern an diesem Netz, sowie eine Musterrealisierung des Netzes an der Genfer Universität, umfasst.

Zieliński K., Indulska J., Krasowski R.: The UMMLAN-2 — a local computer network

INFORMATYKA 1986, No. 11—12, p. 8

Definition of the local network idea, characteristics of the actual standardization work and VLSI technology development for such networks, as well as discussion of hardware architecture and software of the UMMLAN-2 network, which was realized at the University of Mining and Metallurgy in Kraków.

Zieliński K., Indulska J., Krasowski R.: Lokales Computernetz UMMLAN-2

INFORMATYKA 1986, Nr. 11—12, S. 8

Eine Definierung des Begriffes von einem lokalen Computernetz, eine Charakteristik des aktuellen Zustandes von Normungsarbeit und Entwicklung der VLSI-Technologie für einen solchen Netztyp, sowie eine Besprechung der Hardwarearchitektur und Software von dem UMMLAN-2-Netz, das an der Bergbau- und Hüttenakademie in Kraków, realisiert wurde.

Dworakowski W.: IBM PC personal computers (3)

INFORMATYKA 1986, No. 11—12, p. 11

Third part of the IBM PC computers hardware characteristics which includes discussion of the successive, controllers for peripheral devices on the available market.

Dworakowski W.: IBM PC Personal Computer (3)

INFORMATYKA 1986, Nr. 11—12, S. 11

Dritter Teil einer Charakteristik von IBM PC-Hardware, der eine Besprechung von weiteren, auf dem inländischen Markt erreichbaren, Steuerungen für Peripherieanlagen umfasst.

Grabowicz R., Zalewski J.: PC-DOS operating system (3)

INFORMATYKA 1986, No. 11—12, p. 13

Continuation of PC-DOS operating system characteristics, which includes discussion of the remaining commands of the system — version 2.10.

Grabowicz R., Zalewski J.: PC-DOS-Betriebssystem (3)

INFORMATYKA 1986, Nr. 11—12, S. 13

Beendigung einer Charakteristik von dem PC-DOS-Betriebssystem, die eine Besprechung der übrigen Befahlsarten des Systems umfasst.

Maćków P.: Serial communication in the IBM PC and compatibles

INFORMATYKA 1986, No. 11—12, p. 15

Discussion of use and cooperation of existing computer hardware with IBM PC microcomputers and compatibles via a serial interface. Hardware structure of such interface and serial channel software are discussed.

Maćków P.: Serielle Kommunikation im IBM PC und seinen Kompatiblen

INFORMATYKA 1986, Nr. 11—12, S. 15

Eine Besprechung des Problems von Ausnutzung und Zusammenwirken der bestehenden Hardware mit dem IBM PC und seinen Kompatiblen über eine serielle Schnittstelle. Es wurde Hardwarearchitektur solcher Schnittstelle, sowie Software eines seriellen Kanals, besprochen.

Ambroziak J.: Expert systems (1)

INFORMATYKA 1986, No. 11—12, p. 19

First part of expert systems characteristics, which includes discussion of the genesis of such systems and the typical solutions.

Ambroziak J.: Expertensysteme (1)

INFORMATYKA 1986, Nr. 11—12, S. 19
Erster Teil einer Charakteristik von Expertensystemen, der eine Besprechung der Genese von solchen Systemen und ihrer typischen Lösungen umfasst.

Wrycza S.: Premises and conditions of conceptual modelling

INFORMATYKA 1986, No. 11—12, p. 22

Characteristics of the conceptual modelling and it's application in the data base design process. Matter of standardization done in this area by ANSI and ISO are discussed.

Wrycza S.: Voraussetzungen und Bedingen der Konzeptualmodellierung

INFORMATYKA 1986, Nr. 11—12, S. 22

Eine Charakteristik des Wesens von Konzeptualmodellierung und ihrer Anwendung im Datenbankprojektierungsprozess. Es wurde die Thematik, der in diesem Bereich von ANSI und ISO geführten Normungsarbeit besprochen.

Terminologia lokalnych sieci komputerowych (1)

Przy okazji publikowania kilku artykułów na temat lokalnych sieci komputerowych, warto zastanowić się nad terminologią tej dziedziny techniki, stosunkowo młodej i wnoszącej do języka informatyki wiele nowych wyrazów i zwrotów. Choć jest oczywiste, że dziedzina ta nie pozostaje w próżni i jej terminologia opiera się na tradycyjnym słownictwie telekomunikacyjnym i informatycznym, wiele pojęć i terminów jest zupełnie nowych i powstaje w związku z wprowadzeniem nowych rozwiązań technicznych.

Podstawą do wszelkich rozważań technicznych na temat sieci komputerowych jest tzw. **model odniesienia ISO/OSI** (ang. open system interconnection). Wyróżnia się w nim siedem warstw zapewniających łączność na różnych poziomach usług sieciowych. Poszczególne warstwy nazywają się — w kolejności od najniższej do najwyższej:

- **fizyczna** (ang. physical)
- **łącza danych** (ang. data link)
- **sieciowa** (ang. network)
- **transportowa** (ang. transport)
- **sesji** (ang. session)
- **prezentacji** (ang. presentation)
- **aplikacji** (ang. application).

Warstwa fizyczna zapewnia właściwości mechaniczne, elektryczne, funkcjonalne i proceduralne niezbędne do uaktywnienia, utrzymania i anulowania połączeń fizycznych do transmisji bitowej między jednostkami łącza danych. Celem warstwy łącza danych jest zapewnienie środków funkcjonalnych i proceduralnych do ustanowienia, utrzymywania i zwolnienia połączeń między jednostkami sieciowymi. Zadaniem tej warstwy jest wykrywanie i możliwe poprawianie błędów powstałych w warstwie fizycznej. Ponadto, zapewnia ona warstwie sieciowej możliwość zestawienia połączeń (ang. data circuits) w obrębie warstwy fizycznej, tj. możliwość sterowania komutacją łączy (ang. circuit switching).

Pozostałe warstwy modelu ISO/OSI są w chwili obecnej mniej istotne dla lokalnych sieci komputerowych, dlatego ich omówienie można pominąć. Warto jedynie dodać, że w każdej warstwie wyróżnia się jej funkcje wewnętrzne i usługi zapewniane warstwie wyższej. W ten sposób powiązane warstwy tworzą spójny model hierarchiczny, bardzo użyteczny przy projektowaniu i budowie sieci komputerowych.

Opracowane dotąd normy lokalnych sieci komputerowych dotyczą w zasadzie tylko dwóch najniższych warstw modelu ISO, dlatego rozważania terminologiczne można ograniczyć też tylko do tych dwóch warstw.

Najważniejsza z praktycznego punktu widzenia jest obecnie norma IEEE Std. 802.3-1985 (IEEE Standard for Local Area Networks — Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification), wzorowana na sieci Ethernet. Cechą charakterystyczną tej normy jest zdefiniowanie metody współpracy z siecią, tzw. CSMA/CD, polegającej na wspólnym dostępie (ang. multiple access) do ośrodka transmisji (ang. medium), przy zapewnieniu wrażliwości na falę nośną (ang. carrier sense), z wykrywaniem kolizji (ang. collision detection).

W dokumencie „The Ethernet — A Local Area Network Data Link Layer and Physical Layer Specifications" (Version 2.0, November 1982), opublikowanym przez firmy DEC, Intel i Xerox, kolizję definiuje się jako wynik łącznych transmisji zachodzących na siebie w kanale fizycznym, powodujący zniekształcenie danych i konieczność ponownej transmisji. Według specyfikacji Ethernet wrażliwość na falę nośną i wykrywanie kolizji są zapewniane przez warstwę fizyczną, która sygnalizuje odpowiednie stany sygnałami CARRIER_SENSE i COLLISION_DETECT.

CARRIER_SENSE jest to sygnał przekazywany z warstwy fizycznej do warstwy łącza danych, wskazujący że w określonej chwili jedna stacja lub kilka stacji dokonuje transmisji przez kanał.

COLLISION_DETECT jest to sygnał przekazywany z warstwy fizycznej do warstwy łącza danych, wskazujący że jedna lub więcej stacji współzawodniczy w transmisji ze stacją lokalną (może mieć wartość TRUE tylko podczas transmisji).

Przez stację (ang. station) w sieci Ethernet rozumie się pojedyncze adresowalne urządzenie, na ogół zrealizowane jako komputer z odpowiednim wyposażeniem, dołączone do sieci przez sterownik i nadajnik-odbiornik. Sterownik (ang. controller) łączący stację z siecią Ethernet obejmuje część funkcji warstwy fizycznej i większość lub całość funkcji warstwy łącza danych. Nadajnik-odbiornik (ang. transceiver) stanowi część realizacji warstwy fizycznej, dołączoną bezpośrednio do kabla współosiowego i zawiera zarówno elektronikę, służącą do nadawania i odbierania zakodowanych sygnałów, jak i odpowiednią izolację elektryczną.

Sieć lokalna IEEE 802.3 jest również zgodna z modelem ISO/OSI, jednakże struktura warstwy łącza danych i warstwy fizycznej tej sieci jest określona nieco bardziej szczegółowo. Warstwa łącza danych składa się z dwóch podwarstw — LLC (ang. logical link control, sterowanie łączem logicznym) i MAC (ang. medium access control, sterowanie dostępem do ośrodka). Realizacja warstwy fizycznej jest w tej sieci określona jeszcze bardziej konkretnie, mianowicie na warstwę fizyczną składają się następujące elementy:

- **MDI** (ang. medium dependent interface, sprzężenie z ośrodkiem), tj. mechaniczne i elektryczne sprzężenie z kablem
- **MAU** (ang. medium attachment unit, jednostka połączenia z ośrodkiem)
- **AUI** (ang. attachment unit interface, sprzężenie z jednostką połączeniową), tj. kabel, złącza i odpowiednia elektronika
- **PLS** (ang. physical signalling, sygnalizacja fizyczna), tj. wydzielona część warstwy fizycznej zawarta w urządzeniu końcowym transmisji danych (ang. DTE, data terminal equipment), zapewniająca logiczne i funkcjonalne sprzężenie z warstwą łącza danych.

Z powodu podziału warstwy łącza danych na dwie podwarstwy, w normie IEEE 802.3 definicje sygnałów wrażliwości na falę nośną i wykrywania kolizji (w tym wypadku — obecności kolizji, ang. collision presence) różnią się nieco od podanych powyżej.

CARRIER_SENSE jest to sygnał przekazywany z warstwy fizycznej do podwarstwy dostępu do ośrodka, wskazujący że w określonej chwili jedna stacja lub kilka stacji dokonuje transmisji przez kabel współosiowy.

COLLISION_PRESENCE jest to sygnał przekazywany z warstwy fizycznej do podwarstwy dostępu do ośrodka, wskazujący że wiele stacji współzawodniczy w dostępie do ośrodka transmisji.

Przedstawione dane dotyczące struktury modelu ISO/OSI i wynikającej stąd struktury sieci lokalnych, choć nie mają charakteru ściśle terminologicznego, posłużą za punkt odniesienia do dokonania przeglądu nazewnictwa stosowanego w tej dziedzinie, co rozpoczniemy w następnym numerze Informatyki.

JANUSZ ZALEWSKI

KONIEC TWOICH PROBLEMÓW

Nareszcie wszystkie potrzebne Ci dane dotrą na czas w przejrzystej formie tabel i wykresów.

Pakiet oprogramowania na mikrokomputery 16-bitowe

MEGA — BANK

to nowe MEGA możliwości jakie otwierają się przed Twoim przedsiębiorstwem.

Wyobraź sobie

MILIARD

rekordów, które możesz zapełnić według własnych potrzeb i uznania.

Miliard kooperantów, miliard pracowników, miliard produktów — z tym wszystkim nasz MEGA-BANK poradzi sobie bez trudu.

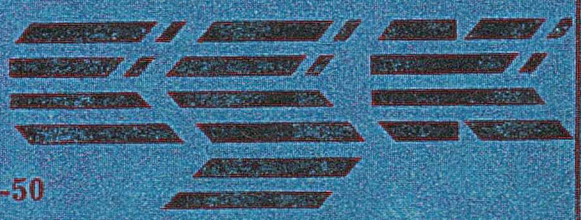
System jest łatwy w obsłudze i opracowany w języku polskim.

Gwarantujemy satysfakcję!

COMPUTER STUDIO KAJKOWSCY

PROFESJONALNE OPROGRAMOWANIE MIKROKOMPUTERÓW

ul. Balladyny 3B, 81-524 Gdynia, tel.: 29-00-18, 24-01-50



EO/BU/K/86